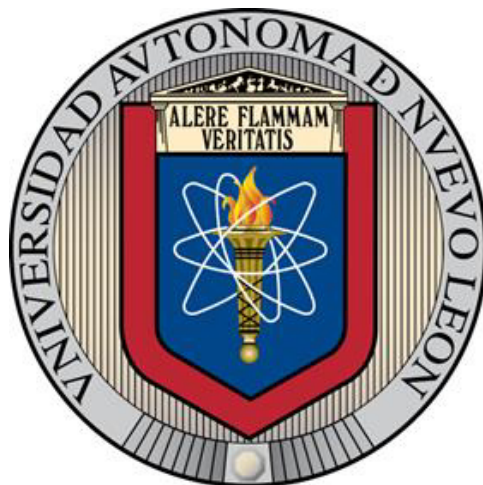


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



TESIS

**MODELOS Y ALGORITMOS DE OPTIMIZACIÓN
PARA LA DISTRIBUCIÓN DE AYUDA
HUMANITARIA**

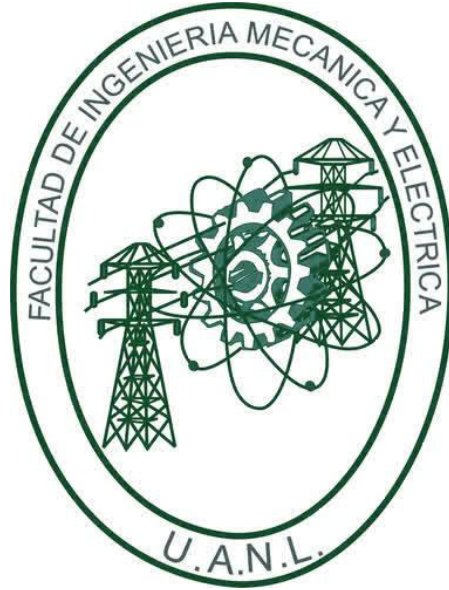
PRESENTADA POR

JOSÉ ÁLVARO GÓMEZ PANTOJA

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
DOCTOR EN INGENIERÍA CON ESPECIALIDAD EN INGENIERÍA
DE SISTEMAS**

ABRIL, 2018

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



TESIS

**MODELOS Y ALGORITMOS DE OPTIMIZACIÓN
PARA LA DISTRIBUCIÓN DE AYUDA
HUMANITARIA**

PRESENTADA POR

JOSÉ ÁLVARO GÓMEZ PANTOJA

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
DOCTOR EN INGENIERÍA CON ESPECIALIDAD EN INGENIERÍA
DE SISTEMAS**

ABRIL, 2018

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis “Modelos y Algoritmos de Optimización para la Distribución de Ayuda Humanitaria”, realizada por el alumno José Álvaro Gómez Pantoja, con número de matrícula 1423780, sea aceptada para su defensa como opción al grado de Doctor en Ingeniería con Especialidad en Ingeniería de Sistemas.

El Comité de Tesis




Dra. María Angélica Salazar Aguilar

Directora



Dr. José Luis González Velarde

Co-Director



Dr. Vincent André Lionel Boyer

Revisor



Dr. Francisco Román Ángel Bello Acosta

Revisor



Dra. Ada Margarita Álvarez Socarrás

Revisor

Vo. Bo.



Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado



San Nicolás de los Garza, Nuevo León, abril 2018

ÍNDICE GENERAL

Agradecimientos	x
Resumen	xii
1. Introducción	1
1.1. Trabajo continuo de ayuda	1
1.2. Alivio de desastres	4
1.3. Metodología de investigación	6
1.4. Estructura de la tesis	7
2. Problema de Asignación de Recursos en un Banco de Alimentos	9
2.1. Revisión de literatura	9
2.2. Descripción del problema	13
2.3. Formulación matemática	15
2.4. Modelo de optimización y complejidad	16
2.5. Resultados computacionales	19

3. Heurística adaptativa	23
3.1. Heurística adaptativa	24
3.2. Operador constructor/reparador	24
3.3. Operadores destructores	30
3.4. Heurística adaptativa vs CPLEX: resultados computacionales	32
4. Método de solución alternativo	35
4.1. Operador constructor/reparador híbrido	36
4.2. Resultados computacionales	38
5. Reformulación del m-CCTP	43
5.1. Descripción del problema	43
5.2. Formulación matemática	44
5.3. Modelo de optimización	45
5.4. Reformulación	47
5.5. Desigualdades válidas	49
5.6. Resultados computacionales	51
6. Conclusiones y trabajo futuro	56
A. Tablas de resultados del FBRAP	58
B. Tablas de resultados del m-CCTP	65
B.1. Punto de referencia para el m-CCTP	65

B.2. Resultados obtenidos en esta tesis para el m-CCTP 70

ÍNDICE DE FIGURAS

2.1. Gap (%) por instancia en clase A.	21
2.2. Gap (%) por instancia en clase B.	21
2.3. Gap (%) por instancia en clase C.	22
3.1. Valor de función objetivo, instancias en clase A	33
3.2. Valor de función objetivo, instancias en clase B	34
3.3. Valor de función objetivo, instancias en clase C	34
4.1. Valor de función objetivo para instancias en clase A.	39
4.2. Valor de función objetivo para instancias en clase B.	39
4.3. Valor de función objetivo para instancias en clase C.	40
5.1. Ruta a seguir por un vehículo.	50

ÍNDICE DE TABLAS

2.1. Características de las instancias de cada clase.	19
2.2. Mejor, promedio, y peor porcentaje de Gap (%) reportado por CPLEX.	22
3.1. Tiempo computacional promedio y Gap (%).	33
4.1. Tiempo computacional para los métodos de solución propuestos. . . .	38
4.2. Número de veces en las que el método reportó mejores valores de la función objetivo.	40
4.3. Brecha relativa entre soluciones (Gap (%)).	41
5.1. Tiempos de llegada a cada nodo de la Figura 5.1.	50
5.2. Soluciones enteras encontradas para cada clase de instancias.	52
5.3. Gap (%) promedio reportado por CPLEX para cada clase de instancias.	53
5.4. Gap (%) promedio entre métodos por clase de instancia.	54
5.5. Comparación puntual entre soluciones.	55
A.1. Resultados computacionales para FBRAP para instancias de clase A.	59
A.2. Resultados computacionales para FBRAP para instancias de clase B.	61

A.3. Resultados computacionales para FBRAP para instancias de clase C.	63
B.1. Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase A.	66
B.2. Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase B.	67
B.3. Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase C.	69
B.4. Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase D.	69
B.5. Tiempo computacional requerido en instancias cuyo óptimo fue al- canzado por CPLEX.	71
B.6. Resultados computacionales obtenidos para instancias en clase A. . .	71
B.7. Resultados computacionales obtenidos para instancias en clase B. . .	72
B.8. Resultados computacionales obtenidos para instancias en clase C. . .	74
B.9. Resultados computacionales obtenidos para instancias en clase D. . .	75

AGRADECIMIENTOS

Agradezco a todas las personas e instituciones que pusieron un granito de arena para la culminación de este trabajo.

A mis padres Isabel Pantoja Cedillo y José Álvaro Gómez Carranza por su apoyo incondicional.

A la Dra. Angélica Salazar, directora de esta tesis, gracias por su confianza, por su apoyo, por estar siempre al pie del cañón. Gracias por creer en mi y por su paciencia...

Al Dr. José Luis González Velarde, codirector de este trabajo, gracias por compartir su sabiduría, tanto en la maestría como en el doctorado, gracias por sus aportaciones.

Al Dr. Vincent André Lionel Boyer, por el apoyo y las observaciones a lo largo de todo el doctorado, gracias además por ser revisor de este trabajo.

Gracias también a la Dra. Ada Margarita Álvarez Socarrás y al Dr. Francisco Román Ángel Bello Acosta por sus enseñanzas de los cursos en los que participé y por ser revisores de esta tesis.

Gracias a la Universidad Autónoma de Nuevo León, a la Facultad de Ingeniería Mecánica y Eléctrica y al Programa de Posgrado en Ingeniería de Sistemas por la oportunidad de cursar el programa doctoral. Gracias también al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca de manutención brindada, gracias

al Programa de Apoyo a la Investigación Científica y Tecnológica (PAICYT-UANL) por el apoyo recibido.

Por último, gracias también a todos mis amigos y compañeros por los momentos vividos a lo largo de estos poco más de tres años.

RESUMEN

José Álvaro Gómez Pantoja.

Candidato para obtener el grado de Doctor en Ingeniería con especialidad en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título: MODELOS Y ALGORITMOS DE OPTIMIZACIÓN PARA LA DISTRIBUCIÓN DE AYUDA HUMANITARIA.

Número de páginas: 80.

OBJETIVO Y METODOLOGÍA DE ESTUDIO: El objetivo general de esta tesis es analizar, modelar y proponer alternativas de solución para dos problemas de logística humanitaria. Estos problemas se relacionan con el trabajo continuo de ayuda y el alivio en desastres.

En lo referente al trabajo continuo de ayuda se introduce el Problema de Asignación de Recursos en un Banco de Alimentos (FBRAP por sus siglas en inglés) y para el alivio en desastres se presenta una reformulación para el Problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP).

El FBRAP es el problema al que se enfrentan día con día en un banco de alimentos. Este problema consta de un conjunto de beneficiarios que demandan ayuda

y un conjunto alimentos disponibles (inventario), así como un presupuesto disponible para adquirir aquéllos productos que permitan asegurar que los donativos son adecuados. Dado que generalmente el Banco de Alimentos no cuenta con una cantidad de productos suficiente para satisfacer las demandas de todos los beneficiarios, el problema de optimización consiste en determinar el subconjunto de beneficiarios que serán apoyados con un donativo, así como en determinar el subconjunto de productos que conformarán dicho donativo. Al configurar los donativos para cada beneficiario, se deben cumplir los requerimientos calóricos necesarios para cumplir con las características de una dieta balanceada, respetando compatibilidades entre beneficiarios y productos, así como diversidad en los productos que integran el donativo.

En esta tesis, se propone un modelo matemático, y dos algoritmos de solución para el FBRAP. El primero de los algoritmos está basado en un esquema ALNS (Adaptive Large Neighbourhood Search) teniendo como único operador reparador un constructivo inspirado en GRASP (Greedy Randomized Adaptive Search Procedure), el segundo es una hibridación entre un algoritmo heurístico ALNS que cuenta con un único operador reparador que asume el modelo propuesto como un problema separable y resuelve cada subproblema con el modelo propuesto. Finalmente se evalúa el desempeño de los métodos de solución propuestos a través de experimentación computacional con un conjunto grande de instancias.

Por otra parte, el m-CCTP consiste en determinar un conjunto de rutas tal que la suma de tiempos de llegada a cada nodo se minimiza (es decir, se busca minimizar la latencia); una de las restricciones principales es que el tiempo de duración de cada ruta no debe exceder un tiempo límite preestablecido. En este problema se consideran tres conjuntos de nodos (obligatorios, opcionales e inalcanzables). Los nodos obligatorios deben ser incluidos en alguna de las rutas; los nodos opcionales, pueden o no ser incluidos en alguna ruta; los nodos inalcanzables que, aunque no pueden incluirse en una ruta, deben de ser cubiertos por al menos uno de los nodos opcionales que han sido elegidos para visitarse.

En este trabajo se propone una reformulación del problema basada en restricciones de flujo, además, se introducen desigualdades válidas a fin de fortalecer dicha reformulación. Se valida y evalúa el alcance de la formulación propuesta, con y sin desigualdades válidas, utilizando el conjunto de instancias evaluadas en la literatura.

CONTRIBUCIONES: Se propone en esta tesis un modelo matemático para el Problema de Asignación de Recursos en un Banco de Alimentos con características que no habían sido estudiadas en la literatura. Se presentan también dos métodos de solución, uno de ellos obtiene soluciones que en promedio están a menos del 6% de la cota superior encontrada para cada instancia y por tanto a menos del 6% del óptimo.

Adicionalmente, se propone una reformulación para el Problema de Recorridos de Cobertura con tiempos Acumulativos, así como un conjunto de desigualdades válidas con el fin de fortalecer la reformulación propuesta. Cabe señalar que con estas reformulaciones se encuentran soluciones de mejor calidad que las reportadas en la literatura, para algunas instancias, además de asegurar optimalidad en algunas de las soluciones encontradas.

Firma de directores:: _____

Dra. María Angélica Salazar Aguilar

Dr. José Luis González Velarde

CAPÍTULO 1

INTRODUCCIÓN

Dentro del ámbito de logística humanitaria se distinguen dos principales ramificaciones, por un lado el trabajo continuo de ayuda y por otro los temas relacionados a alivio en desastres (ver Kovács y Spens [21]). Aunque en ocasiones el alivio en desastres cubre parte de la lucha contra el hambre (ver Long [25]) normalmente el término alivio en desastres se maneja para catástrofes inesperadas tales como desastres naturales y desastres provocados por el hombre.

Con el fin de abarcar ambas ramas, en esta tesis se estudia un problema de cada rama, con el fin de identificar sus características y proponer modelos y algoritmos para su solución: el Problema de Asignación de Recursos en un Banco de Alimentos y el Problema de Recorridos de Cobertura con Tiempos Acumulativos.

1.1 TRABAJO CONTINUO DE AYUDA

El trabajo continuo de ayuda se conforma por actividades a largo plazo, incluso permanentes, con el fin de apoyar al sustento de la población en cuanto al cuidado de la salud, vivienda y alimentos. En esta tesis se aborda el Problema de Asignación de Recursos en un Banco de Alimentos (FBRAP por sus siglas en inglés), el cual se centra en la ayuda permanente a la sociedad sin tomar en cuenta la ayuda en desas-

tres, siendo entonces una contribución al trabajo continuo de ayuda (ver Venkatesh et al. [36]).

Cada año, la población mundial aumenta y con la meta de cubrir sus necesidades se desencadena una serie de efectos negativos como la deforestación del suelo que disminuye la capacidad para cultivar alimentos, el precio de los productos alimenticios se torna inestable, etc., como uno de los resultados de estos efectos tenemos un crecimiento alarmante en la cantidad de personas que no tienen acceso a alimentos adecuados y balanceados.

Hay varios trabajos en la literatura que contienen cifras alarmantes sobre la malnutrición y la pobreza alimentaria, lo que representa un área de oportunidad. Según la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO), entre 2012 y 2014, al menos 805 millones de personas en todo el mundo sufrieron hambre y escasez de alimentos. Más aún, una de cada siete personas no tiene una dieta que cumpla con sus requerimientos energéticos (ver FAO et al. [11]).

Para reducir el hambre en el mundo se requiere la mejora de varios factores clave, entre ellos, aumentar la productividad agrícola, mejorar el acceso a los suministros de alimentos y la gestión/utilización de los mismos, desarrollar las zonas rurales, implementar programas específicos de nutrición, etc. (ver FAO et al. [11]). Teniendo en cuenta todos estos factores, se tiene una gran oportunidad para aplicar investigación de operaciones a fin de ayudar al tomador de decisiones en turno a diseñar/configurar una asignación eficiente de recursos. La motivación para este trabajo se centra en ayudar a la toma de decisiones en una organización sin fines de lucro, en este caso, un banco de alimentos.

Un banco de alimentos es una organización sin fines de lucro que recibe donativos de productos alimenticios y en algunos casos dinero (ver González-Torre y Coque [18]) y los dirige a los sectores necesitados de la población. Por lo general, los donativos de alimentos son recuperados de las diferentes etapas de la cadena alimentaria (productores, procesadores, minoristas y consumidores) con el fin de evitar que

se conviertan en desperdicio. Dichos alimentos pueden surgir de diferentes fuentes, como pedidos cancelados, productos que llegaron a su fecha de consumo preferente y siguen siendo comestibles pero no comercializables; productos que tienen baja demanda, productos con empaque dañado pero que aún son aptos para consumo, cantidades incorrectas de pedidos, etc. (ver Alexander y Smaje [2]). Por otro lado, las donaciones monetarias tienen en la mayoría de los casos propósitos fiscales.

Los bancos de alimentos juegan entonces un papel importante en la lucha contra el hambre. Según Wilson y Tsoa [37], en marzo de 2001, el 2.4 % de las personas en Canadá recibieron asistencia de emergencia por parte de algún banco de alimentos. En marzo de 2002, la Asociación Canadiense de Bancos de Alimentos informó que su programa de alimentos de emergencia tenía 747,665 usuarios de los cuales 305,000 de ellos eran niños menores de 18 años. Además, en 2003, 3 millones de personas en Canadá vivían en hogares con inseguridad alimentaria, lo cual representaba 10.2 % de los hogares canadienses (ver McIntyre [29]). Además, el informe End Hunger Report 2013 [4] muestra que el 41 % de las agencias de beneficencia en Australia obtuvieron alimentos de los bancos de alimentos una vez a la semana y otro 41 % recibió donativos una o dos veces al mes.

Por lo tanto, entre las contribuciones de este trabajo podemos encontrar, por un lado, la recuperación y redirección de una gran cantidad de productos alimenticios que de otro modo se perderían a lo largo de toda la cadena alimentaria (recolección, manejo, comercialización, consumo) y, por otro lado, la ayuda humanitaria, que ataca directamente la vulnerabilidad alimenticia en la que se encuentra un gran número de personas, teniendo así una importante contribución social.

En el Problema de Asignación de Recursos en un Banco de Alimentos se tiene un conjunto de beneficiarios que demandan ayuda, sin embargo, en general no se tienen recursos suficientes para cubrir todas las demandas y por tanto se tiene que decidir a qué beneficiarios se les asignará un donativo. Cada beneficiario tiene una demanda y una prioridad que refleja el grado de vulnerabilidad en el que se encuentra,

y para configurar su donativo, en caso de que sea seleccionado, se deben tomar en cuenta restricciones de manejo de inventario y compras, además de restricciones que aseguren que el donativo cumple con las características de una dieta balanceada, así como compatibilidades entre productos y beneficiarios.

En este trabajo se modela el FBRAP como un problema lineal entero mixto y se proponen dos algoritmos de solución. El primero es una heurística adaptativa que está basada en los mecanismos principales de las metaheurísticas ALNS (Adaptive Large Neighborhood Search) y GRASP (Greedy Randomized Adaptive Search Procedure), el segundo explota las características del problema con una colaboración entre un método de solución exacto y algunos de los mecanismos de la metaheurística ALNS.

Se seleccionaron las metaheurísticas GRASP y ALNS como base para los métodos de solución propuestos porque han presentado un excelente desempeño al resolver problemas combinatorios. GRASP es una metaheurística propuesta por Feo y Resende [12] que consiste en un generador de soluciones iniciales y una fase de mejora. GRASP se ha utilizado ampliamente para resolver problemas combinatorios complejos, como los problemas de programación (scheduling) [1, 32] y los problemas de ruteo de vehículos [7, 26], entre otros. Por otra parte ALNS es una poderosa metaheurística propuesta por Ropke y Pisinger [33], cuya principal ventaja es el proceso adaptativo que guía la búsqueda en función de las características de la instancia en cuestión. Finalmente se evalúa el desempeño de los métodos de solución propuestos a través de experimentación computacional sobre un conjunto grande de instancias.

1.2 ALIVIO DE DESASTRES

La segunda rama de la logística humanitaria está relacionada con el alivio de desastres, dichos desastres pueden ser naturales tales como: huracanes, terremotos, tornados, temblores, erupciones volcánicas, etc., o bien provocados por el hombre

como lo pueden ser accidentes nucleares y ataques relacionados con el terrorismo.

El manejo de desastres se puede dividir según Lee y Zbinden [23] en tres fases de operaciones:

- **Operaciones de preparación:** Su fin es reducir el impacto ocasionado por un desastre en caso de que suceda.
- **Operaciones durante el desastre:** Son operaciones de respuesta inmediata al desastre, tales como planes de emergencia.
- **Operaciones post desastre:** Son operaciones de reconstrucción, contribuyen a estabilizar el entorno tanto en el ámbito social como de infraestructura.

En esta tesis se hace también una contribución a la segunda rama de la logística humanitaria relacionada con el alivio en desastres, específicamente a la fase de operaciones durante el desastre. Se presenta una reformulación al Problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP por sus siglas en inglés), introducido por Flores-Garza et al. [14], dicho problema se enfoca en las actividades a corto plazo, centradas en brindar productos de primera necesidad de manera inmediata después de la ocurrencia de un desastre.

El Problema de Recorridos de Cobertura con Tiempos Acumulativos consiste en determinar un conjunto de rutas tal que la suma de tiempos de llegada a cada localidad es minimizada (es decir, se minimiza la latencia), una de las restricciones es que el tiempo de duración de cada ruta no debe exceder un tiempo límite preestablecido. En este problema se consideran tres conjuntos de nodos (obligatorios, opcionales e inalcanzables). Los nodos obligatorios deben ser incluidos en alguna de las rutas; los nodos opcionales, pueden o no ser incluidos en alguna ruta; y los nodos inalcanzables que, aunque no pueden incluirse en una ruta, deben de ser cubiertos por al menos uno de los nodos opcionales que han sido elegidos para visitarse. Para este problema, se propone una reformulación del problema basada en restricciones

de flujo en redes, además de fortalecerla con desigualdades válidas. El alcance y validación de la reformulación propuesta, se realiza utilizando el conjunto de instancias analizadas por Flores-Garza et al. [14].

1.3 METODOLOGÍA DE INVESTIGACIÓN

La metodología seguida para el trabajo de tesis presentado consistió en los siguientes pasos:

Para el Problema de Asignación de Recursos en un Banco de Alimentos (FBRAP):

- Revisión de literatura relacionada con Bancos de Alimentos y logística humanitaria.
- Formulación matemática para el FBRAP.
- Generación de instancias.
- Solución de instancias mediante el optimizador CPLEX en su versión 12.6 con el fin de validar el modelo y probar su alcance.
- Desarrollo e implementación computacional de un algoritmo heurístico.
- Desarrollo e implementación computacional de un algoritmo conformado por una componente heurística y una componente exacta.
- Experimentación computacional de los métodos de solución propuestos sobre las instancias generadas.
- Análisis de resultados y comparación de métodos de solución.
- Escritura de la tesis.

Para el Problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP):

- Reformulación matemática del m-CCTP basada en restricciones de flujo en redes.
- Fortalecimiento de la reformulación propuesta con desigualdades válidas.
- Solución de instancias mediante el optimizador CPLEX en su versión 12.6 con el fin de validar la formulación y probar su alcance.
- Análisis de resultados obtenidos.
- Escritura de la tesis.

1.4 ESTRUCTURA DE LA TESIS

La estructura de la tesis es la siguiente: en el Capítulo 2 se presenta una descripción del Problema de Asignación de Recursos en un Banco de alimentos (FBRAP), se discute el estado del arte del problema propuesto, se introduce una formulación matemática del mismo y se presentan resultados computacionales.

En el Capítulo 3 se presenta un algoritmo heurístico basado en las componentes principales de las metaheurísticas ALNS y GRASP para resolver el FBRAP, además de un análisis de los resultados obtenidos por dicho método de solución.

En el Capítulo 4 se presenta un método de solución alternativo para el FBRAP, el cual está basado en la combinación de componentes de una metaheurística (ALNS) y un método de solución exacto, se presentan resultados computacionales y se comparan con los métodos de solución anteriormente presentados en los Capítulos 2 y 3.

En el Capítulo 5 se presenta una reformulación del Problema de Recorridos de Cobertura con tiempos Acumulativos, así como un conjunto de desigualdades válidas que fortalecen la formulación propuesta. Se discuten también los resultados computacionales obtenidos.

Finalmente en el Capítulo 6 se presentan las conclusiones de la tesis y el trabajo futuro.

CAPÍTULO 2

PROBLEMA DE ASIGNACIÓN DE RECURSOS EN UN BANCO DE ALIMENTOS

Existen en la literatura algunos trabajos que abordan el problema al que se enfrentan diariamente los bancos de alimentos, sin embargo, no se conoce algún trabajo que haya estudiado el problema bajo las mismas características que se consideran en el problema abordado en esta tesis.

En este capítulo se presenta el estado del arte del Problema de Asignación de Recursos en un Banco de Alimentos (FBRAP), una descripción formal del problema, así como el modelo de optimización propuesto. Por último se presenta una discusión de los resultados computacionales obtenidos.

2.1 REVISIÓN DE LITERATURA

En la literatura, la mayoría de los trabajos relacionados con los problemas de bancos de alimentos se centran en una de las siguientes áreas: ubicación de depósitos o centros de distribución, pronóstico de demanda o donaciones y ruteo de vehículos

para el flujo de productos, ya sea en la etapa de recolección (donantes a centros de distribución y/o beneficiarios) o en la etapa de distribución (depósitos a beneficiarios). La característica común de estos trabajos es que sus objetivos se centran en mejorar factores como el tiempo de viaje, la igualdad entre los beneficiarios, la sostenibilidad, la eficiencia, etc.

En el trabajo presentado por Davis et al. [9] se estudia la programación de la recolección de productos desde la ubicación de los donadores así como la programación de la entrega de los productos recolectados a agencias de caridad ya sea directamente del donador, o bien, utilizando puntos de entrega, que sirven como puntos de conexión entre donadores y destinatarios. Presentan un modelo de cobertura de conjuntos utilizado para asignar agencias de caridad a cada lugar de entrega de alimentos. Entre otras cosas también se consideran las limitaciones de la capacidad del vehículo, la frecuencia de las visitas a los donadores y los horarios de trabajo.

Una variante del problema de ruteo/ubicación fue introducida por Solak et al. [35], en el que consideran un almacén central desde donde se entregan los alimentos a las agencias de caridad. La decisión es una decisión táctica, que consiste en la asignación de beneficiarios a un conjunto de sitios de entrega, y también la programación de las rutas a seguir por los vehículos de entrega. De la misma manera que en Davis et al. [9] se maneja un solo producto. Cabe mencionar que en el problema que se estudia en esta tesis no hay la necesidad de tomar tales decisiones, dado que cada beneficiario seleccionado para recibir donativo debe dirigirse al banco de alimentos para obtenerlo.

Un estudio relacionado con el pronóstico de donativos recibidos en un banco de alimentos está dado por Brock y Davis [6], en el cual se predice la cantidad de productos alimenticios que se recolectarán de los supermercados en un horizonte de planeación dado, esto con el objetivo de evitar costos adicionales de inventario y/o transporte. Se evaluaron cuatro métodos diferentes de pronóstico y el mejor pronóstico fue el reportado por una red neuronal de perceptrón multicapa.

En el trabajo propuesto por Orgut et al. [31] se considera un problema de asignación de recursos que tiene en cuenta dos características importantes: i) un flujo continuo de productos, que es una característica clave para marcar la diferencia entre una cadena de desastres y una cadena estable a largo plazo; y ii) el caso real que enfrentan las organizaciones caritativas donde, en la mayoría de los casos, los recursos recaudados son insuficientes para satisfacer la demanda total de todos los beneficiarios. Su medida de desempeño se basa en la igualdad de beneficiarios.

En esta tesis se aborda el problema de la asignación de recursos en los bancos de alimentos como una cadena desarrollada a largo plazo que contempla los casos donde el suministro no puede satisfacer la demanda y una de las principales contribuciones y diferencias con el trabajo de Orgut et al. [31] es considerar no un flujo homogéneo, sino una variedad heterogénea de productos alimenticios.

La Cumbre Mundial de la Alimentación (World Food Summit [38]) establece que “La seguridad alimentaria existe cuando todas las personas, en todo momento, tienen acceso físico y económico a alimentos suficientes, seguros y nutritivos que cumplan sus necesidades nutricionales y preferencias alimenticias para una vida activa y sana” y según Barrett [5] la seguridad alimentaria se puede medir por medio de cuatro dimensiones. La primera dimensión es la disponibilidad de alimentos, que incluye no solo la cantidad de alimentos disponibles, sino también la calidad y diversidad de los mismos. La segunda dimensión evalúa el acceso a los alimentos, incluye indicadores de acceso físico y la infraestructura que tienen los beneficiarios para el adecuado almacén y manejo de los productos. La tercera dimensión verifica la estabilidad e incluye factores que miden el riesgo de exposición a la inseguridad alimentaria, los precios volátiles de los alimentos, etc. La última dimensión, pero no menos importante, es la utilización, la cual evalúa el uso correcto y las ventajas de los productos alimenticios. El trabajo presentado por FAO et al. [11] considera solo una parte de la tercera dimensión (estabilidad).

Por lo tanto, para tener en cuenta, en la medida de lo posible, las cuatro dimen-

siones anteriores, en este trabajo se incluyen algunas restricciones para garantizar una dieta equilibrada [17, 22, 30], y un uso adecuado de los productos alimenticios (por ejemplo, la leche infantil debe asignarse a un beneficiario en donde existan bebés). Además, este trabajo permite priorizar a los beneficiarios acorde al grado de vulnerabilidad en el que se encuentran.

Otro trabajo en una organización sin fines de lucro es el propuesto por Lien et al. [24], en el que se presenta un problema de asignación de recursos secuencial con servicio equitativo. Los autores definen el servicio en términos de la relación entre el monto asignado y la demanda observada, y la función objetivo es maximizar la tasa de cumplimiento mínima esperada entre los clientes. Es importante resaltar que no toman en cuenta productos heterogéneos. Un caso similar se puede encontrar en Guo et al. [19].

En el trabajo de Martins et al. [27] se estudia un problema de distribución de alimentos en un banco de alimentos. El problema consiste en asignar productos donados a todos los beneficiarios que demandan ayuda sin favorecer a alguno en particular, y el objetivo es maximizar la satisfacción global de nutrientes en las organizaciones, asegurando que la relación entre el donativo y la necesidad de cada nutriente sea similar en todos los beneficiarios. Este problema también tiene en cuenta productos con características nutricionales similares y productos que pueden donarse solo a un cierto tipo de beneficiarios. Las principales diferencias del trabajo presentado en esta tesis son las siguientes: se consideran diferentes prioridades (grados de necesidad) de los beneficiarios, se establecen condiciones para que la configuración de un donativo cumpla con especificaciones de una dieta equilibrada y la consideración de un presupuesto para la compra de productos.

2.2 DESCRIPCIÓN DEL PROBLEMA

Dado un conjunto de beneficiarios N y un conjunto de productos M , el problema consiste en determinar el subconjunto de beneficiarios que recibirán donativos, así como el conjunto de productos que conformarán dicho donativo. Nótese que el donativo debe cumplir con los mínimos requerimientos calóricos de una dieta balanceada y respetar las compatibilidades entre beneficiarios y productos. Cada beneficiario $n \in N$, que puede estar conformado por un único individuo, una familia, una población o una organización (iglesia, hospital, escuela, casa hogar, asilo, etc.), tiene una demanda ideal en Calorías (D_n) para satisfacer sus requerimientos alimenticios y un grado de vulnerabilidad (prioridad, necesidad) W_n , el cual ha sido determinado por medio de un estudio social.

Cuando los recursos del banco de alimentos (inventario de productos o presupuesto para compras) no son suficientes para satisfacer las necesidades de todos los beneficiarios utilizamos las diferentes prioridades para llevar a cabo la asignación de donativos. Una de las medidas más usuales para calcular el grado de vulnerabilidad (prioridad) de los beneficiarios es el Índice de Vulnerabilidad Social (SVI por sus siglas en inglés) el cual, según Flanagan et al. [13], consta de los siguientes cuatro rubros:

- Estatus socioeconómico: Estadísticas de desempleo, pobreza, ingreso per cápita, educación y servicios de salud.
- Composición del hogar: Edad, hogar monoparental, capacidades diferentes, etc.
- Minorías: Idioma, etnias, etc.
- Vivienda/Transporte: número de individuos en una casa, disponibilidad de vehículos, etc.

Dado que cada beneficiario puede tener diferentes características y requerimien-

tos alimenticios, para calcular su demanda ideal se pueden tomar en cuenta puntos como la cantidad de individuos que conforman al beneficiario, así como información de cada uno de ellos: requerimiento energético, género, edad, tamaño, composición corporal (ver Food and Agriculture Organization of the United Nations [15]), días que cubrirá el donativo, y otras particularidades (lactancia, maternidad, (ver McGuire [28])). Adicionalmente, el contenido total de Calorías T_m de cada producto $m \in M$ así como las calorías correspondientes a carbohidratos C_m , proteínas P_m , y grasas L_m son conocidas de antemano. Los valores energéticos usados para la conversión peso-Calorías fueron basados en el Sistema Atwater [3] y son 4 Cal/g para carbohidratos y proteínas y 9 Cal/g para grasas, el contenido total de cada producto es calculado como la suma de las calorías correspondientes de cada uno de los tres macronutrientes abordados.

Cada banco de alimentos puede tener diferentes políticas para recibir productos y repartirlos entre los beneficiarios, el caso que motiva este trabajo es un banco de alimentos local en el que un beneficiario $n \in N$ se considera servido si la cantidad asignada de Calorías alcanza un porcentaje mínimo, llamado α_n , de la demanda total D_n , y la cantidad donada cumple con las características de una dieta balanceada. Es decir, se encuentra entre los requisitos mínimo y máximo para cada uno de los nutrientes (carbohidratos, proteínas y grasas). Además, cada producto se clasifica en una de las ocho clases de productos (productos de origen animal, cereales, azúcares, grasas, frutas, verduras, especias y legumbres) y cada beneficiario servido debe recibir productos de al menos β diferentes clases de productos. Si los requisitos anteriores no se cumplen, el beneficiario se considera no atendido.

A continuación se presenta una formulación matemática para representar el Problema de Asignación de Recursos en un Banco de Alimentos (FBRAP).

2.3 FORMULACIÓN MATEMÁTICA

Conjuntos

N : Conjunto de beneficiarios.

M : Conjunto de productos.

K : Conjunto de clases de productos.

Parámetros

D_n Demanda en Calorías del beneficiario $n \in N$.

W_n Peso (prioridad) del beneficiario $n \in N$.

$\alpha_n \in [0.20, 1.0]$ Mínimo porcentaje de demanda para considerar que el beneficiario n es servido.

I_m Inventario del producto $m \in M$.

E_m Precio de compra del producto $m \in M$.

B Presupuesto disponible para compras (\$).

T_m Contenido total de Calorías del producto $m \in M$.

C_m Calorías provenientes de carbohidratos en el producto $m \in M$.

P_m Calorías provenientes de proteínas en el producto $m \in M$.

L_m Calorías provenientes de grasas en el producto $m \in M$.

$[\underline{C}, \bar{C}]$ Proporción mínima y máxima de Calorías relacionadas con carbohidratos requeridas en una dieta balanceada.

$[\underline{P}, \bar{P}]$ Proporción mínima y máxima de Calorías relacionadas con proteínas requeridas en una dieta balanceada.

$[\underline{L}, \bar{L}]$ Proporción mínima y máxima de Calorías relacionadas con grasas requeridas en una dieta balanceada.

β Mínimo número de clases de productos requeridos para servir a algún beneficiario.

$$F_{mk} = \begin{cases} 1 & \text{si el producto } m \in M \text{ pertenece a la clase de productos } k \in K, \\ 0 & \text{en otro caso.} \end{cases}$$

$$G_{nm} = \begin{cases} 1 & \text{si el producto } m \in M \text{ es compatible con el beneficiario } n \in N, \\ 0 & \text{en otro caso.} \end{cases}$$

El parámetro G_{nm} es un auxiliar primordial para evitar asignaciones producto-beneficiario no deseadas, estas asignaciones pueden tener diferentes índoles tales como nutricional (fórmula para lactantes debe ser donada a beneficiarios que tengan infantes, ya que de otro modo sería desperdiciada), culturales (creencias religiosas acerca del manejo de algunos alimentos de origen animal), logísticos (beneficiarios que no poseen la infraestructura mínima para garantizar el correcto transporte/almacén de algunos productos), etc.

VARIABLES DE DECISIÓN

x_{nm} : unidades del producto $m \in M$ asignadas al beneficiario $n \in N$.

Es importante recalcar que, por la definición del parámetro de compatibilidad G_{nm} , si el producto $m \in M$ no es compatible con el beneficiario $n \in N$, la asignación x_{nm} es fijada en cero, esto reduce la cantidad de variables tomadas en cuenta por el optimizador.

$$y_n = \begin{cases} 1 & \text{si el beneficiario } n \in N \text{ es servido,} \\ 0 & \text{en otro caso.} \end{cases}$$

$$v_{nk} = \begin{cases} 1 & \text{si algún producto de la clase de productos } k \in K \text{ es asignado} \\ & \text{al beneficiario } n \in N, \\ 0 & \text{en otro caso.} \end{cases}$$

w_m : unidades compradas del producto $m \in M$.

2.4 MODELO DE OPTIMIZACIÓN Y COMPLEJIDAD

$$\text{(FBRAP) Maximizar } Z = \sum_{n=1}^{|N|} W_n y_n \tag{2.1}$$

subject to

$$\alpha_n D_n y_n \leq \sum_{m=1}^{|M|} T_m x_{nm} \leq D_n y_n \quad n \in N \quad (2.2)$$

$$\underline{C} \sum_{m=1}^{|M|} T_m x_{nm} \leq \sum_{m=1}^{|M|} C_m x_{nm} \leq \bar{C} \sum_{m=1}^{|M|} T_m x_{nm} \quad n \in N \quad (2.3)$$

$$\underline{P} \sum_{m=1}^{|M|} T_m x_{nm} \leq \sum_{m=1}^{|M|} P_m x_{nm} \leq \bar{P} \sum_{m=1}^{|M|} T_m x_{nm} \quad n \in N \quad (2.4)$$

$$\underline{L} \sum_{m=1}^{|M|} T_m x_{nm} \leq \sum_{m=1}^{|M|} L_m x_{nm} \leq \bar{L} \sum_{m=1}^{|M|} T_m x_{nm} \quad n \in N \quad (2.5)$$

$$\sum_{n=1}^{|N|} x_{nm} \leq I_m + w_m \quad m \in M \quad (2.6)$$

$$\sum_{m=1}^{|M|} E_m w_m \leq B \quad (2.7)$$

$$v_{nk} \leq \sum_{m=1}^{|M|} F_{mk} x_{nm} \quad n \in N, k \in K \quad (2.8)$$

$$\sum_{k=1}^{|K|} v_{nk} \geq \beta y_n \quad n \in N \quad (2.9)$$

$$x_{nm} \in \mathbb{Z}^+ \quad n \in N, m \in M \quad (2.10)$$

$$y_n \in \{0, 1\} \quad n \in N \quad (2.11)$$

$$v_{nk} \in \{0, 1\} \quad n \in N, k \in K \quad (2.12)$$

$$w_m \in \mathbb{Z}^+ \quad m \in M \quad (2.13)$$

Considerando el grado de vulnerabilidad (prioridad) como puntuación, la función objetivo (2.1) (suma ponderada) busca maximizar el puntaje total recolectado con los beneficiarios servidos. Las restricciones (2.2) son utilizadas para identificar cuáles de los beneficiarios deben ser apoyados con un donativo. Las restricciones (2.3)–(2.5) aseguran que se cumplen los requerimientos mínimos y máximos para carbohidratos, proteínas y grasas, respectivamente, esto para cumplir con las condiciones de una dieta balanceada. Las restricciones (2.6) establecen la condición de que la asignación total no sobrepase las unidades de inventario aunadas a las compras realizadas para cada uno de los productos. Por su parte la restricción (2.7) asegura que las compras que realice el banco de alimentos no excedan el presupuesto que

se tiene disponible. Las restricciones (2.8) se encargan de identificar de qué familias provienen los productos que fueron usados para configurar el donativo de cada beneficiario servido, y así utilizar las variables auxiliares v_{nk} para asegurar que se cumple con el mínimo requerimiento de clases de productos (2.9). Las restricciones restantes denotan la naturaleza de las variables.

Es importante resaltar que los valores mínimos y máximos de las proporciones de cada uno de los macronutrientes, así como el valor de β puede depender de las características y necesidades de cada uno de los beneficiarios (ver especificaciones en [28]). Sin embargo, en este trabajo se ha fijado el valor de β en 5 y las proporciones mínimas y máximas de Calorías requeridas por cada uno de los macronutrientes manejados (carbohidratos, proteínas y grasas) en $[0.60, 0.70]$, $[0.10, 0.15]$ y $[0.20, 0.30]$, respectivamente. Además, si no hubiera prioridad en los beneficiarios $W_n = 1 \forall n \in N$.

Un caso simple del problema abordado es cuando el banco de alimentos no tiene disponible un presupuesto para realizar compras de productos ($B = 0$), el mínimo número de clases de productos para servir a los beneficiarios es cero ($\beta = 0$), y las proporciones mínimas y máximas de Calorías correspondientes a cada nutriente son fijadas en cero y uno (*i.e.* $\underline{C} = \underline{P} = \underline{L} = 0$ y $\bar{C} = \bar{P} = \bar{L} = 1$), respectivamente. Bajo estas condiciones el problema se reduce a: función objetivo (2.1), restricciones (2.2), naturaleza de variables (2.10), (2.11) y la restricción (2.6) queda reformulada como:

$$\sum_{n=1}^{|N|} x_{nm} \leq I_m \quad m \in M. \quad (2.14)$$

Por lo tanto, el problema resultante puede verse como una variante de dos problemas combinatorios bien conocidos: el problema de la mochila múltiple con restricciones de asignación (MKAR) que es NP-hard [8, 10] y el problema de la suma de subconjuntos también NP-hard ([16]). Este hecho nos da como resultado que los problemas mencionados son al menos tan difíciles de resolver como el problema abordado en este trabajo.

2.5 RESULTADOS COMPUTACIONALES

Al realizar la revisión de literatura no se encontraron trabajos previos que estudien el Problema de Asignación de Recursos en un Banco de Alimentos (FBRAP) con las características aquí consideradas. Por lo tanto, se presentan resultados computacionales para la validación de la formulación propuesta, considerando un gran conjunto de instancias. Para llevar a cabo dicha experimentación, se implementó el modelo en lenguaje C++ y se resolvieron las instancias con el optimizador CPLEX en su versión 12.6 en una CPU Intel Xeon(R) a 2.10 GHz E5-2620 v2 bajo el sistema operativo Ubuntu 14.04 LTS. Los criterios de paro para cada una de las instancias fueron dos horas de tiempo de cómputo (7200 s) y/o un Gap de 0.01 %.

Se generaron aleatoriamente tres clases de instancias A, B y C, con 50 instancias cada una, dicha generación se realizó basándose en una base de datos de productos. Además, con base en las políticas de un banco de alimentos local, se generaron al azar la prioridad de cada beneficiario, el inventario existente para cada producto, el presupuesto disponible y la compatibilidad entre los productos y beneficiarios. Es importante mencionar que cada producto tiene múltiples unidades existentes además de que el número mínimo de diferentes clases de productos asignados a un beneficiario servido se estableció en cinco, y el requisito mínimo α_n para considerar un beneficiario como servido se estableció en 0.20 de su demanda total. Los tamaños de las instancias generadas se muestran en la Tabla 2.1.

Tabla 2.1: Características de las instancias de cada clase.

Clase	Beneficiarios	Productos
A	100	200
B	300	200
C	500	200

Con el fin de evaluar la formulación propuesta (FBRAP), se trató de resolver

cada instancia con el optimizador CPLEX en su versión 12.6 y con el fin de reducir el espacio de búsqueda y mejorar los resultados computacionales se añadió la siguiente desigualdad válida:

$$\sum_{m=1}^{|M|} F_{mk} x_{nm} \leq \bar{M}_n v_{nk} \quad n \in N, k \in K \quad (2.15)$$

donde \bar{M}_n representa una cota superior para la cantidad de productos que se pueden donar al beneficiario n . El valor de \bar{M}_n es calibrado con:

$$\bar{M}_n = \frac{D_n}{\min_{m \in M} \{T_m\}} \quad (2.16)$$

Las Figuras 2.1-2.3 muestran el Gap (%) reportado por CPLEX. En la Figura 2.1 se puede ver que aunque el Gap (%) arrojado por CPLEX para cada una de las instancias de la clase A es muy pequeño, ninguna de las instancias fue resuelta a optimalidad, es decir, en todas las instancias el optimizador paró al llegar al tiempo límite preestablecido (7200 segundos). Por otro lado, CPLEX reporta un Gap (%) más grande en las clases de instancias B y C (ver Figuras 2.2 y 2.3), que en algunos casos excede el 100%, esto se debe a la forma en que el Gap es calculado:

$$Gap(\%) = 100 \times \left[\frac{\text{Mejor Cota Superior} - \text{Mejor Solución Encontrada}}{\text{Mejor Solución Encontrada}} \right] \quad (2.17)$$

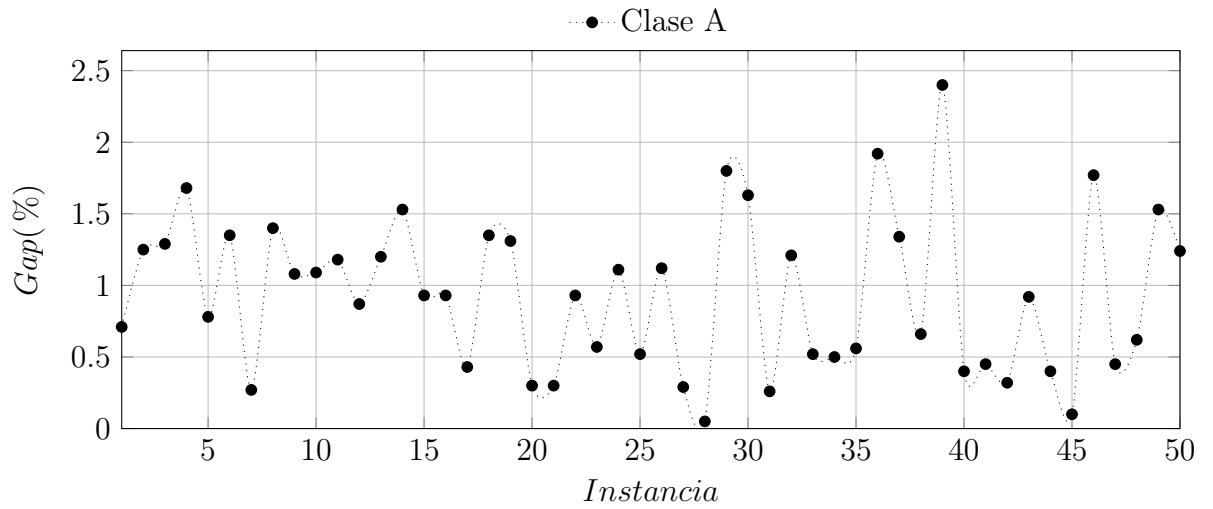


Figura 2.1: Gap (%) por instancia en clase A.

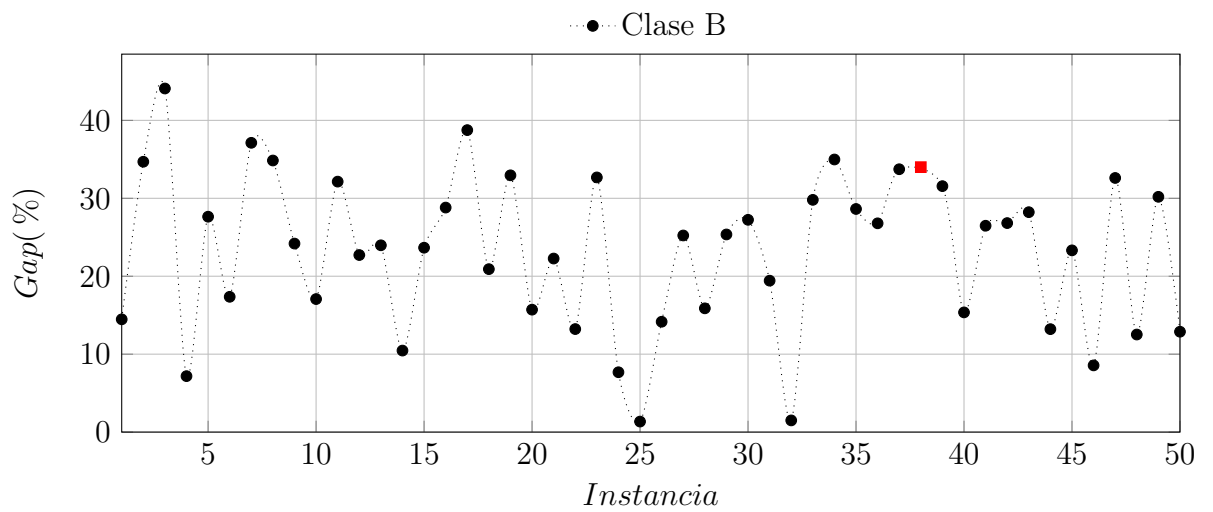


Figura 2.2: Gap (%) por instancia en clase B.

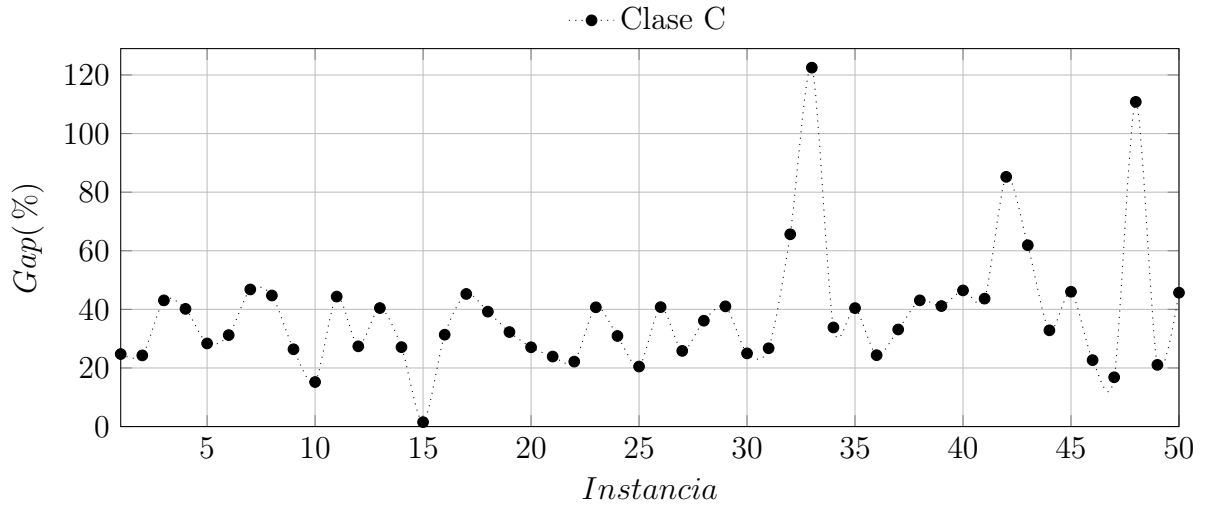


Figura 2.3: Gap (%) por instancia en clase C.

Note que en la Figura 2.2 el Gap (%) correspondiente a la instancia número 38 tiene una marca alternativa debido a que el valor real fue removido con el fin de tener una mejor apreciación de los resultados, el valor correspondiente es 1045.81 %.

En la Tabla 2.2 se muestran el mejor, el peor y el promedio de los porcentajes de Gap reportados por CPLEX para las instancias de cada una de las clases.

Tabla 2.2: Mejor, promedio, y peor porcentaje de Gap (%) reportado por CPLEX.

Clase	A	B	C
Mínimo	0.05	1.35	1.51
Promedio	0.94	43.52	38.24
Máximo	2.40	1045.81	122.50

Cabe mencionar que aunque CPLEX encuentra soluciones de muy buena calidad para las instancias de la clase A, su desempeño se ve comprometido con el incremento del número de beneficiarios (clases B y C), alcanzando el criterio de paro del tiempo límite (2 horas) sin obtener una solución de buena calidad (ver Tabla 2.2), por tal motivo es necesario el diseño e implementación de otros métodos de solución.

CAPÍTULO 3

HEURÍSTICA ADAPTATIVA

Como se discutió en el Capítulo 2, el alcance del optimizador CPLEX es muy corto en instancias con más de 300 beneficiarios (clases B y C), por tanto surge la necesidad de desarrollo de un método de solución más eficiente. Se propone entonces una heurística adaptativa basada en componentes de las metaheurísticas GRASP y ALNS.

GRASP es una metaheurística propuesta por Feo y Resende [12] que consiste en un generador de de soluciones iniciales y una fase de mejora. ALNS es una poderosa metaheurística propuesta por Ropke y Pisinger [33], cuya principal ventaja es el proceso adaptativo de búsqueda. Se seleccionaron las metaheurísticas GRASP y ALNS como base para los métodos de solución propuestos por su buen desempeño en problemas combinatorios.

En este capítulo se presenta una heurística adaptativa para resolver el Problema de Asignación de Recursos en un Banco de Alimentos, se describen también los operadores constructores y destructores propios del esquema basado en el ALNS y se analizan los resultados computacionales, haciendo comparaciones pertinentes con los resultados computacionales obtenidos en el Capítulo 2.

3.1 HEURÍSTICA ADAPTATIVA

El método de solución propuesto es un algoritmo basado en el mecanismo principal de la metaheurística ALNS en el que hay un conjunto de operadores de destrucción y reparación que intentan mejorar una solución inicial dada (ver Algoritmo 1). En cada iteración se elige un operador destructor y un operador reparador con una probabilidad que es proporcional a su rendimiento en las iteraciones pasadas. El proceso se repite hasta que se cumple el criterio de paro establecido y se reporta la mejor solución encontrada.

En esta heurística adaptativa, se diseñaron cinco operadores de destrucción y todos ellos destruyen (remueven) al menos uno de los beneficiarios pertenecientes a la solución en cuestión. Sólo hay un operador de reparación, que también es encargado de generar la solución inicial. Este operador reparador se basa en un GRASP y en cada iteración, se elige aleatoriamente un parámetro γ y se utiliza para calcular algunos límites inferiores relacionados con la cantidad de Calorías por nutriente que debe asignarse a los beneficiarios seleccionados. La descripción detallada del ALNS propuesto se brinda en las siguientes secciones.

3.2 OPERADOR CONSTRUCTOR/REPARADOR

Como se mencionó anteriormente, el operador utilizado para reparar una solución parcialmente destruida y el generador de la solución inicial son los mismos, por lo tanto, este procedimiento funciona tanto a partir de una solución vacía como de una solución parcial (ver Algoritmo 2). Supongamos que $Y \subset N$ es el conjunto de beneficiarios incluidos en la solución actual S_c y que $T \subset N \setminus Y$ sea el conjunto de beneficiarios etiquetados como tabú (es decir, los beneficiarios para los cuales no fue posible configurar un donativo factible, o bien, los beneficiarios que fueron removidos de la solución por algún operador de destrucción). Supongamos que $H \leq B$ es la

Algoritmo 1: Heurística Adaptativa

```

 $S_c \leftarrow \emptyset$  (Solución actual);
 $Y \leftarrow \emptyset$  (Beneficiarios servidos);
 $T \leftarrow \emptyset$  (Conjunto de beneficiarios con etiqueta tabú);
 $S_c \leftarrow \text{OperadorReparador}(S_c, N, Y, T)$  (Construir una solución inicial) ;
 $S^* \leftarrow S_c$  ;
para  $i=0::3$  hacer
    Remove aleatoriamente el  $i \times 10\%$  de los beneficiarios servidos en  $S_c$  ;
     $\triangleright$ Diversificación
     $T \leftarrow$  etiquetar como tabú los beneficiarios que fueron removidos en el paso anterior;
    para  $j=1::5$  hacer
        Elegir operador destructor  $\delta$ ;
         $T \leftarrow T \cup \delta(S_c)$ ;
         $S_c \leftarrow \text{OperadorReparador}(S_c, N, Y, T)$ ;
        si  $S_c$  es mejor que  $S^*$  entonces
             $S^* \leftarrow S_c$ ;
            Habilitar todos los operadores;
             $j = 1$ ;
        en otro caso
             $S_c \leftarrow S^*$ ;
            si  $\delta \neq D4$  entonces
                Etiquetar al operador  $\delta$  como tabú;
            fin
             $j=j+1$ ;
        fin
     $T \leftarrow \emptyset$ ;
fin
devolver  $S^*$ 

```

cantidad de dinero que se ha utilizado para comprar productos, entonces $B - H$ es el presupuesto que se tiene disponible para seguir realizando compras. Observe que, $Y = \emptyset$, $T = \emptyset$, y $H = 0$ si $S_c = \emptyset$.

Algoritmo 2: Operador Reparador

Datos: S_c, N, Y, T

$U \leftarrow N \setminus \{Y \cup T\}$;

repetir

$\bar{n} \leftarrow \arg \max_{n \in U} W_n / D_n$;

 Elegir γ para calcular $DC_{\bar{n}}, DP_{\bar{n}}, DL_{\bar{n}}$;

 (Paso 1.) Asignar al beneficiario \bar{n} , si es posible, una unidad de β diferentes

 productos que contienen el menor contenido calórico y contribuyen con β clases de productos (*ActualizarAsignación*(1, \bar{m}, \bar{n})) ;

repetir

 Construir *LRC*;

 Seleccionar aleatoriamente el producto \bar{m} a asignar;

 Determinar la cantidad de unidades ($x_{\bar{n}\bar{m}}$) del producto m que serán asignadas al beneficiario \bar{n} (*ActualizarAsignación*($x_{\bar{n}\bar{m}}, \bar{m}, \bar{n}$)) ;

hasta que la asignación de productos al beneficiario \bar{n} ha alcanzado las cotas dadas por (3.1)–(3.3) o *LRC* = \emptyset ;

si X_n no es factible **entonces**

DestruirAsignación(X_n);

$T \leftarrow T \cup \{\bar{n}\}$;

en otro caso

$Y \leftarrow Y \cup \{\bar{n}\}$;

 Actualizar U, S_c , etc.;

fin

hasta que $U = \emptyset$;

devolver S_c

En cada una de las iteraciones, el algoritmo identifica a los beneficiarios que no han sido servidos y crea el conjunto de beneficiarios candidatos $U \subset N \setminus \{Y \cup T\}$, luego se elige el beneficiario \bar{n} que tiene el valor más alto de utilidad, es decir, $\bar{n} = \arg \max_{n \in U} \{W_n / D_n\}$. Después de eso, se calcula el límite inferior para cada uno de los nutrientes de la siguiente manera:

$$DC_{\bar{n}} = \frac{\underline{C} + \gamma(\bar{C} - \underline{C})}{(1 - \gamma)(\underline{C} + \underline{P} + \underline{L}) + \gamma(\bar{C} + \bar{P} + \bar{L})} \times \alpha_{\bar{n}} \times D_{\bar{n}} \quad (3.1)$$

$$DP_{\bar{n}} = \frac{\underline{P} + \gamma(\bar{P} - \underline{P})}{(1 - \gamma)(\underline{C} + \underline{P} + \underline{L}) + \gamma(\bar{C} + \bar{P} + \bar{L})} \times \alpha_{\bar{n}} \times D_{\bar{n}} \quad (3.2)$$

$$DL_{\bar{n}} = \frac{\underline{L} + \gamma(\bar{L} - \underline{L})}{(1 - \gamma)(\underline{C} + \underline{P} + \underline{L}) + \gamma(\bar{C} + \bar{P} + \bar{L})} \times \alpha_{\bar{n}} \times D_{\bar{n}} \quad (3.3)$$

donde $DC_{\bar{n}}$, $DP_{\bar{n}}$ y $DL_{\bar{n}}$ representan la cantidad mínima de calorías de carbohidratos, proteínas y grasas, que deben asignarse al beneficiario \bar{n} para que pueda considerarse como servido. Observe que el valor de γ se elige aleatoriamente de $\Gamma = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ y su probabilidad de elección (ρ_γ) está determinada por su peso ω_γ , el cual a su vez está asociado con su desempeño en iteraciones pasadas. La primera vez que se llama al operador de reparación, todos los $\gamma \in \Gamma$ tienen el mismo peso (es decir, $\omega_\gamma = 1/|\Gamma|$) y en los siguientes pasos, el peso y la probabilidad de selección se actualizan de la siguiente manera:

$$\omega_\gamma = (S(\gamma) + 1) / (U(\gamma) + 1)$$

$$\rho_\gamma = \omega_\gamma / \sum_{\theta \in \Gamma} \omega_\theta$$

donde $U(\gamma)$ es el número de veces que se ha seleccionado el valor de $U\gamma$ y $S(\gamma)$ es la cantidad de veces que este valor ha permitido asignaciones factibles de productos alimenticios a los beneficiarios, esta fórmula fue adaptada de Ropke y Pisinger [33] y fue usada en el trabajo de Salazar-Aguilar et al. [34].

Una vez que se ha elegido el beneficiario candidato a servir y se han calculado los límites inferiores correspondientes a cada uno de los nutrientes, se crea una lista de productos candidatos LC . Cada producto $m \in LC$ debe ser compatible con el beneficiario \bar{n} y $I_m > 0$ o $B - H \geq E_m$ (es decir, debe haber al menos una unidad de m en el inventario o presupuesto suficiente para comprarlo). Entonces, el operador

reparador propuesto considera tres pasos principales para llevar a cabo la asignación de productos al beneficiario \bar{n} .

- *Paso 1:* Asignar al beneficiario \bar{n} , si es posible, una unidad de β productos diferentes de LC que contengan la menor cantidad de Calorías y contribuyan con β nuevas clases de productos. Hay dos casos posibles: el primer caso es cuando $|LC| < \beta$ o la cantidad de clases de las que provienen los productos es inferior a β , bajo estas características no se puede realizar ninguna asignación factible; y el segundo caso es cuando $|LC| \geq \beta$ y la cantidad de clases de las cuales provienen los productos es de al menos β . Si ocurre el segundo caso, todos los productos en LC se ordenan en orden decreciente de acuerdo con su contenido total de Calorías y el algoritmo selecciona una unidad de β productos diferentes que contienen la menor cantidad de Calorías y contribuyen con β clases de productos (es decir, seleccione una unidad de los primeros beta productos que provienen de diferentes clases de productos). Si se puede realizar una asignación factible se va al *Paso 2*, de lo contrario, se va a *Paso 4*. Este paso garantiza que se satisfaga la variedad de clases de productos (ver restricción (2.9)).
- *Paso 2:* Construir una lista restringida de candidatos (LRC) de la LC , seleccionar un producto $\bar{m} \in LRC$ y determinar la cantidad de unidades ($x_{\bar{n}\bar{m}}$) que debe asignarse al beneficiario \bar{n} . Con el fin de construir la LRC , el algoritmo calcula las cotas inferiores (requerimientos) por nutriente ($DC_{\bar{n}}, DP_{\bar{n}}, DL_{\bar{n}}$), necesarias para cubrir las necesidades alimentarias del beneficiario \bar{n} ; luego identifica el nutriente que está mas distante de su cota inferior, *i.e.* si los índices $\{1, 2, 3\}$ corresponden a $\{C, P, L\}$ y $n_1 = DC_{\bar{n}}, n_2 = DP_{\bar{n}}, n_3 = DL_{\bar{n}}$, entonces el nutriente (i^*) con la demanda más grande a ser cumplida está dado por $i^* = \operatorname{argmax}_{i \in \{1, 2, 3\}} \{n_i\}$.

Después, el algoritmo calcula la contribucion de cada producto $m \in LC$, y solamente el 10% de los productos con la mejor contribución del nutriente i^* son incluidos en la LRC . Luego, los productos en LRC se ordenan siguiendo

un orden decreciente del valor de la siguiente función:

$$\bar{e}_m = \frac{\min \{C_m, DC_{\bar{n}}\} + \min \{P_m, DP_{\bar{n}}\} + \min \{L_m, DL_{\bar{n}}\}}{T_m} \quad (3.4)$$

Esta función mide la tasa de contribución de Calorías. Véase por ejemplo el caso de los carbohidratos; si el producto m tiene C_m Calorías de carbohidratos y la demanda actual (límite inferior) de carbohidratos del beneficiario \bar{n} es $DC_{\bar{n}}$, entonces, si $C_m > DC_{\bar{n}}$, sólo $DC_{\bar{n}}$ Calorías se tomarán en cuenta en la función de evaluación, de lo contrario se considerarán C_m Calorías.

Después de ordenar los productos en la LRC , se elige aleatoriamente un producto $\bar{m} \in LRC$ utilizando una distribución de probabilidad triangular. Por ejemplo, si LRC tiene cardinalidad l el i -ésimo producto tiene una probabilidad $\frac{2 \times (l + 1 - i)}{l \times (l + 1)}$ de ser elegido y la cantidad de unidades de producto \bar{m} que se asignará al beneficiario \bar{n} se determina utilizando la siguiente fórmula:

$$x_{\bar{m}\bar{n}} = \max \left\{ 1, \min \left\{ I_{\bar{m}} + \left\lfloor \frac{B - H}{E_{\bar{m}}} \right\rfloor, \left\lfloor \frac{DC_{\bar{n}}}{C_{\bar{m}}} \right\rfloor, \left\lfloor \frac{DP_{\bar{n}}}{P_{\bar{m}}} \right\rfloor, \left\lfloor \frac{DL_{\bar{n}}}{L_{\bar{m}}} \right\rfloor \right\} \right\} \quad (3.5)$$

Una vez que se ha determinado la asignación de productos al beneficiario en cuestión ($x_{\bar{m}\bar{n}}$), el inventario de productos, las compras y los gastos son actualizados de acuerdo al Algoritmo 3 :

- *Paso 3:* Repetir *Paso 2* hasta que la asignación de productos al beneficiario \bar{n} ha alcanzado las cotas inferiores provistas por (3.1)–(3.3) o $LRC = \emptyset$.
- *Paso 4:* Sea $X_{\bar{n}}$ la asignación de productos al beneficiario \bar{n} .
 - Si $X_{\bar{n}}$ no es factible, entonces destruir la asignación de productos y regresar las unidades de producto al Inventario respectivo, o bien, descontar el costo del gasto actual (H), siguiendo los pasos del Algoritmo 4. Posteriormente, añadir el beneficiario \bar{n} al conjunto tabú T .
 - En otro caso, añadir el beneficiario \bar{n} al conjunto de beneficiarios servidos (*i.e.* $Y = Y \cup \{\bar{n}\}$) y actualizar U , S_c , etc.

El operador reparador continúa con la siguiente iteración y así sucesivamente. Se detiene cuando ya no es posible realizar asignaciones factibles y reporta una solución factible S_c .

Una vez que se ha creado una solución factible, la heurística adaptativa intenta *diversificar la búsqueda* eliminando un porcentaje de los beneficiarios servidos en la solución actual. Luego, la solución parcial se reconstruye con el operador de reparación.

Algoritmo 3: Actualizar Asignación.

Datos: $x_{\bar{m}\bar{n}}, \bar{m}, \bar{n}$

si $x_{\bar{m}\bar{n}} \geq I_{\bar{m}}$ **entonces**

$I_{\bar{m}} \leftarrow 0;$	▷Unidades del Inventario
$w_{\bar{m}} \leftarrow w_{\bar{m}} + (x_{\bar{m}\bar{n}} - I_{\bar{m}});$	▷Unidades compradas
$H \leftarrow H + (x_{\bar{m}\bar{n}} - I_{\bar{m}}) \times E_{\bar{m}};$	▷Gastos

en otro caso

$I_{\bar{m}} \leftarrow I_{\bar{m}} - x_{\bar{m}\bar{n}};$	▷Unidades del Inventario
--	--------------------------

fin

$DC_{\bar{n}} \leftarrow \max \{DC_{\bar{n}} - x_{\bar{m}\bar{n}} \times C_{\bar{m}}, 0\};$ ▷Actualización de cotas inferiores de cada nutriente

$DP_{\bar{n}} \leftarrow \max \{DP_{\bar{n}} - x_{\bar{m}\bar{n}} \times P_{\bar{m}}, 0\};$

$DL_{\bar{n}} \leftarrow \max \{DL_{\bar{n}} - x_{\bar{m}\bar{n}} \times L_{\bar{m}}, 0\};$

3.3 OPERADORES DESTRUCTORES

Inicialmente se tienen disponibles cinco operadores destructores, cada uno de los primeros cuatro tiene la misma probabilidad de elección. Es importante resaltar que el quinto operador es usado con el fin de diversificar la búsqueda una vez que se ha encontrado un óptimo local con los otros operadores. Luego, en cada iteración, un operador de destrucción $\delta \in \{D1, D2, D3, D4\}$ se elige aleatoriamente y su probabilidad de selección se determina por su historial de rendimiento (del mismo modo que la selección de γ).

Algoritmo 4: Destruir Asignación.**Datos:** X_n **para** $m=1:|M|$ **hacer** **si** $x_{m\bar{n}} > 0$ **entonces** **si** $w_m \geq x_{m\bar{n}}$ **entonces** $w_m \leftarrow w_m - x_{m\bar{n}} ;$ ▷Unidades compradas $H \leftarrow H - (x_{m\bar{n}} \times E_m) ;$ ▷Gastos **en otro caso** $w_m \leftarrow 0;$ ▷Unidades compradas $H \leftarrow H - (w_m \times E_m) ;$ ▷Gastos $I_m \leftarrow I_m + (x_{m\bar{n}} - w_m) ;$ ▷Unidades del Inventario **fin** **fin****fin**

- *D1. Remover el beneficiario con la demanda más alta (D_n).*
- *D2. Remover el beneficiario con el peor valor de utilidad ($R_n = W_n/D_n$).*
- *D3. Remover el beneficiario con la menor prioridad (W_n).*
- *D4. Remover un beneficiario seleccionándolo aleatoriamente.*

Una vez que el operador destructor δ ha sido seleccionado, la solución actual S_c es destruida parcialmente, de acuerdo con lo contemplado en δ . El beneficiario que acaba de ser removido es marcado con una etiqueta tabú y la solución resultante es reparada por el operador reparador. Si la solución arrojada por el operador reparador es mejor que S_c , esta solución reemplaza a S_c y todos los operadores son marcados como operadores disponibles. En otro caso, si $\delta \neq D4$, entonces el operador δ es etiquetado como tabú. La probabilidad de selección de cada operador destructor disponible es actualizada y el proceso es repetido hasta que consecutivamente se han generado cinco soluciones sin obtener una mejora. Si S_c es una mejor solución que S^* (la mejor solución encontrada hasta el momento), S^* es reemplazada por S_c .

Después de eso, el algoritmo diversifica la búsqueda con el 5º operador destructor (*i.e.* remueve aleatoriamente un porcentaje de beneficiarios servidos de la solución actual S_c) y entonces reconstruye la solución resultante usando el operador reparador. Después de eso, la nueva solución es destruida y reparada iterativamente con el fin de mejorar su calidad. Nótese que aunque la diversificación de la solución se lleva a cabo cuatro veces, la primera vez el operador destructor destruye el 0% de la solución, esto con el fin de explotar la estructura de la solución inicial, entonces se tiene realmente una diversificación en tres pasos; la primera vez un 10% de los beneficiarios atendidos se eliminan de la solución en cuestión, la segunda vez el 20% y la tercera vez el 30%. Finalmente, el algoritmo propuesto reporta la mejor solución encontrada S^* .

3.4 HEURÍSTICA ADAPTATIVA VS CPLEX:

RESULTADOS COMPUTACIONALES

Con el fin de comparar el desempeño del método propuesto se corrió 10 veces la heurística adaptativa con cada una de las instancias de prueba. Luego, se identificó la mejor solución encontrada y el promedio de las soluciones reportadas por la heurística adaptativa en las 10 corridas. Los valores de las funciones objetivo de estas soluciones se compararon con la mejor solución entera reportada por CPLEX y se calculó el Gap (%) entre ellas. La Tabla 3.1 muestra que en las clases B y C la heurística propuesta obtuvo mejores soluciones que CPLEX con un porcentaje promedio de Gap superior al 21%. En las últimas dos filas, se puede observar el número de instancias en las que la heurística y CPLEX obtuvieron mejores soluciones, es decir, para las instancias de la clase B, la heurística superó a CPLEX en 32 de las 50 instancias probadas y, por ejemplo, en la clase C, CPLEX alcanzó mejores soluciones sólo en un caso (ver Figuras (3.1)-(3.3)).

Tabla 3.1: Tiempo computacional promedio y Gap (%).

Clase		A	B	C
Tiempo (s)	Mejor Sol.	104.56	752.75	1225.10
	Prom. Sol.	93.98	584.98	1089.56
Gap (%)	Mejor Sol.	16.41	-21.76	-24.07
	Prom. Sol.	18.16	-20.07	-22.83
Mejores soluciones	CPLEX	50/50	18/50	1/50
	Heurística	0/50	32/50	49/50

El Gap (%) reportado en Tabla 3.1 es calculado de la siguiente manera:

$$Gap(\%) = 100 \times \left[\frac{Z_C - Z_{Hu}}{Z_C} \right]. \quad (3.6)$$

Donde Z_C es la mejor solución reportada por CPLEX y Z_{Hu} es la mejor solución encontrada por la heurística adaptativa (ver Tabla A.1 en Apéndice A). Si el Gap es negativo significa que la heurística reportó mejores soluciones que el optimizador CPLEX.

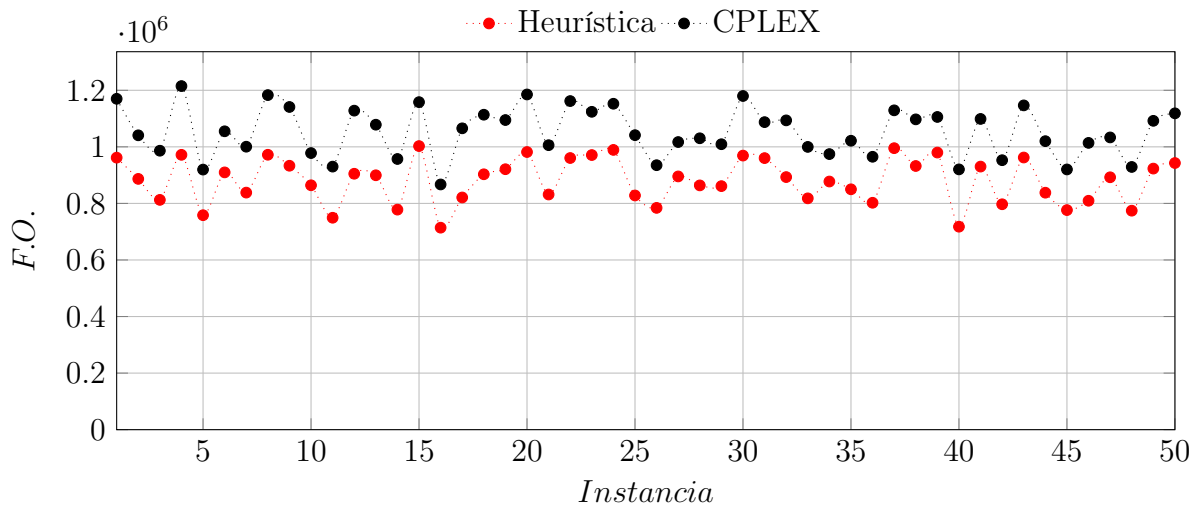


Figura 3.1: Valor de función objetivo, instancias en clase A

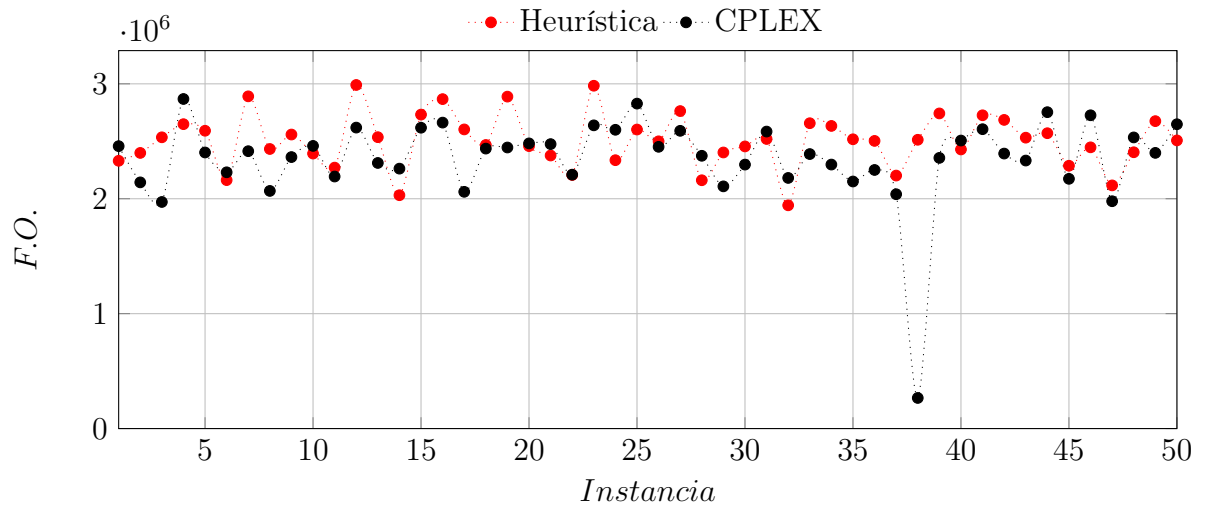


Figura 3.2: Valor de función objetivo, instancias en clase B

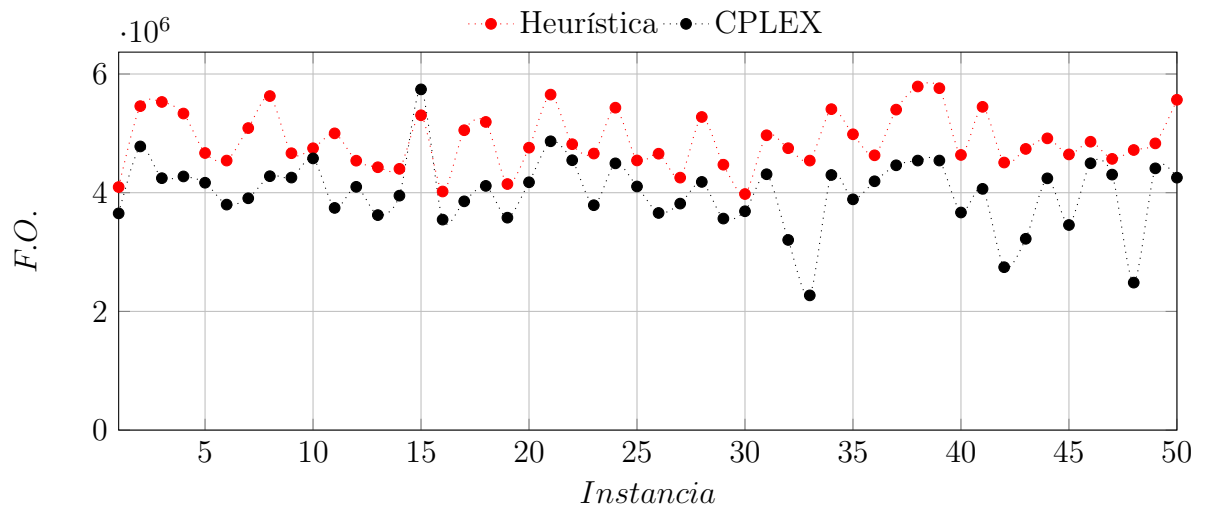


Figura 3.3: Valor de función objetivo, instancias en clase C

CAPÍTULO 4

MÉTODO DE SOLUCIÓN ALTERNATIVO

Dado que el FBRAP es un problema NP-hard, el desempeño de CPLEX se ve afectado negativamente con el incremento en el tamaño de las instancias, sin embargo, reporta soluciones de muy buena calidad cuando la instancia es pequeña (ver resultados para la clase A en la Tabla A.1 del Apéndice). Este comportamiento es utilizado por el operador constructor propuesto en este capítulo, con el fin de mejorar los resultados encontrados con la heurística propuesta en el capítulo anterior.

Dicho lo anterior, con el fin de explotar las fortalezas del modelo matemático presentado en el Capítulo 2 y del método de solución propuesto en el Capítulo 3, se propone un método de solución alternativo. En este método se considera como base la heurística adaptativa del Capítulo 3 y se sustituye el operador reparador, por un operador de reparación/construcción alternativo.

En este capítulo se introduce un operador reparador basado en el modelo matemático del FBRAP, se presenta la actualización de la heurística adaptativa del capítulo anterior y se presentan y analizan resultados computacionales comparando los tres métodos de solución propuestos.

4.1 OPERADOR CONSTRUCTOR/REPARADOR HÍBRIDO

Al igual que el operador constructor/reparador presentado en el capítulo anterior, este nuevo operador se puede utilizar tanto para construir una solución inicial como para reconstruir una solución parcial.

Supongamos nuevamente que $Y \subset N$ es el conjunto de beneficiarios incluidos en la solución actual S_c y que $T \subset N \setminus Y$ son los beneficiarios etiquetados como tabú. Es decir, los beneficiarios para los cuales no fue posible configurar un donativo factible, o bien, los beneficiarios que fueron removidos luego de ser seleccionados por un operador de destrucción. Supongamos que $H \leq B$ es la cantidad de dinero que se ha utilizado para comprar productos. Observe que, $Y = \emptyset$, $T = \emptyset$, y $H = 0$ si $S_c = \emptyset$.

El operador reparador alternativo queda de la manera siguiente:

Algoritmo 5: Operador reparador alternativo

Datos: $S_c, N, Y, T, B - H$

$U \leftarrow N \setminus \{Y \cup T\};$

repetir

$\bar{n} \leftarrow \arg \max_{n \in U} W_n / D_n;$

$X_{\bar{n}} \leftarrow AP(\bar{n}, I, B - H);$

si $y_{\bar{n}} = 1$ **entonces**

$Y \leftarrow Y \cup \{\bar{n}\};$

para $m=1:|M|$ **hacer**

 | *Actualizar Asignación*($x_{\bar{n}m}, \bar{n}, m$)

fin

 Update U, S_c , etc.;

en otro caso

$T \leftarrow T \cup \{\bar{n}\};$

fin

hasta que $U = \emptyset;$

devolver S_c

Este operador, de la misma manera que el operador reparador descrito en el

Algoritmo 2, identifica a los beneficiarios que no han sido servidos y crea el conjunto de beneficiarios candidatos $U \subset N \setminus \{Y \cup T\}$, luego elige el beneficiario \bar{n} que tenga el valor más alto de utilidad (es decir, $\bar{n} = \arg \max_{n \in U} W_n/D_n$) para verificar si se puede configurar un donativo de manera factible.

En contraste con el operador constructor de la sección anterior, este algoritmo en lugar de realizar asignaciones producto-beneficiario de manera heurística invoca al solver CPLEX para resolver de manera exacta el problema dado por restricciones (2.2-2.13) con la desigualdad válida (2.15), considerando la siguiente función objetivo:

$$(AP) \quad \text{Maximizar } Z=y_{\bar{n}} \tag{4.1}$$

Nótese que este es un sub-problema del problema original con un único beneficiario, de tal modo que la nueva función objetivo es la variable binaria relacionada con el beneficiario \bar{n} que se activará solamente en caso que se pueda configurar un donativo $X_{\bar{n}} = \{x_{\bar{n}1}, x_{\bar{n}2}, \dots, x_{\bar{n}m}\}$ factible. Este problema es resuelto iterativamente con cada uno de los beneficiarios candidatos y se realiza una actualización de datos globales (inventarios, gastos, etc...) de acuerdo con el Algoritmo 3, cada que $y_{\bar{n}} = 1$. En caso de que no sea posible configurar un donativo factible, el beneficiario es añadido al conjunto de beneficiarios tabú. El criterio de paro del optimizador para cada sub-instancia es un tiempo límite de un segundo o un Gap de 0%.

Para evaluar el desempeño del algoritmo propuesto se realizaron 10 corridas para cada una de las instancias, de ellas se tomó la mejor solución reportada. Los resultados computacionales, así como las comparativas correspondientes entre métodos de solución se presentan a continuación.

4.2 RESULTADOS COMPUTACIONALES

Con el fin de evaluar y comparar los métodos de solución propuestos para el Problema de Asignación de Recursos en un Banco de Alimentos se presentan a continuación los siguientes resultados. Para el método heurístico y para el método alternativo (híbrido) se realizaron 10 corridas en cada una de las instancias de prueba. En la Tabla 4.1 se reportan los tiempos computacionales para la mejor solución encontrada en cada instancia. Cabe mencionar que el tiempo computacional requerido por el método exacto, al usar la formulación original, se ha omitido debido a que CPLEX en cada una de las instancias llegó al criterio de paro de 7200 segundos.

Tabla 4.1: Tiempo computacional para los métodos de solución propuestos.

	Clase	A	B	C
Heurística	Promedio	104.57	752.75	1225.11
	Mínimo	46.07	264.36	553.36
	Máximo	226.06	1426.19	3263.68
Híbrido	Promedio	11.31	67.17	103.47
	Mínimo	6.17	25.76	30.67
	Máximo	22.99	103.60	208.57

Como se puede ver en la Tabla 4.1 los dos métodos propuestos requieren un menor tiempo computacional que CPLEX, sin embargo, el método alternativo requiere solamente, en promedio, 10 % del tiempo computacional requerido por la heurística.

En las Figuras 4.1-4.3 se muestran los valores de función objetivo por instancia. En la clase A CPLEX mantiene un mejor desempeño que los métodos restantes, sin embargo, CPLEX queda en último lugar con respecto al valor de la función objetivo en las clases de instancias B y C.

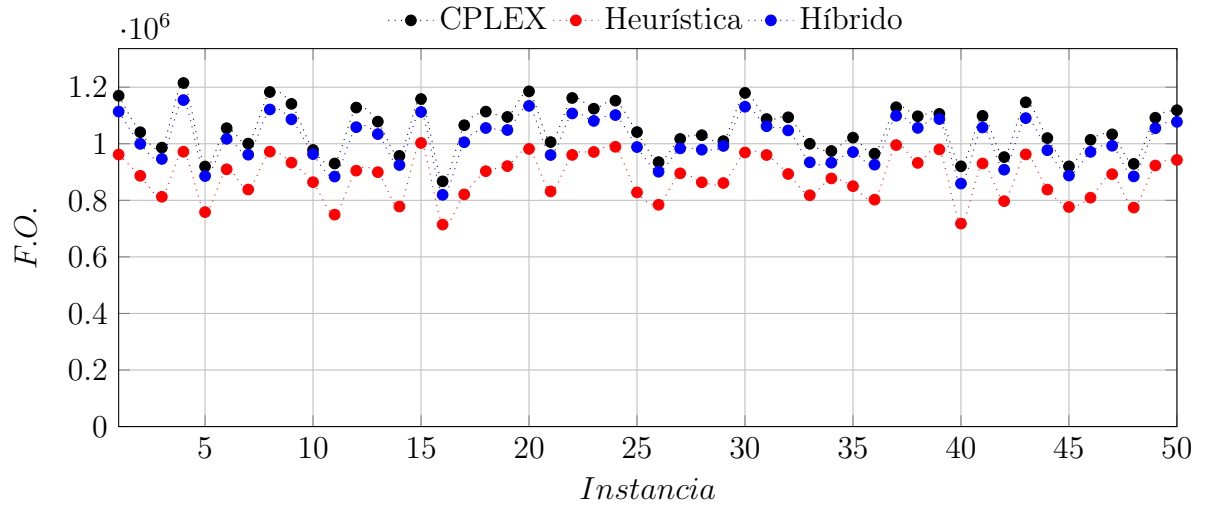


Figura 4.1: Valor de función objetivo para instancias en clase A.

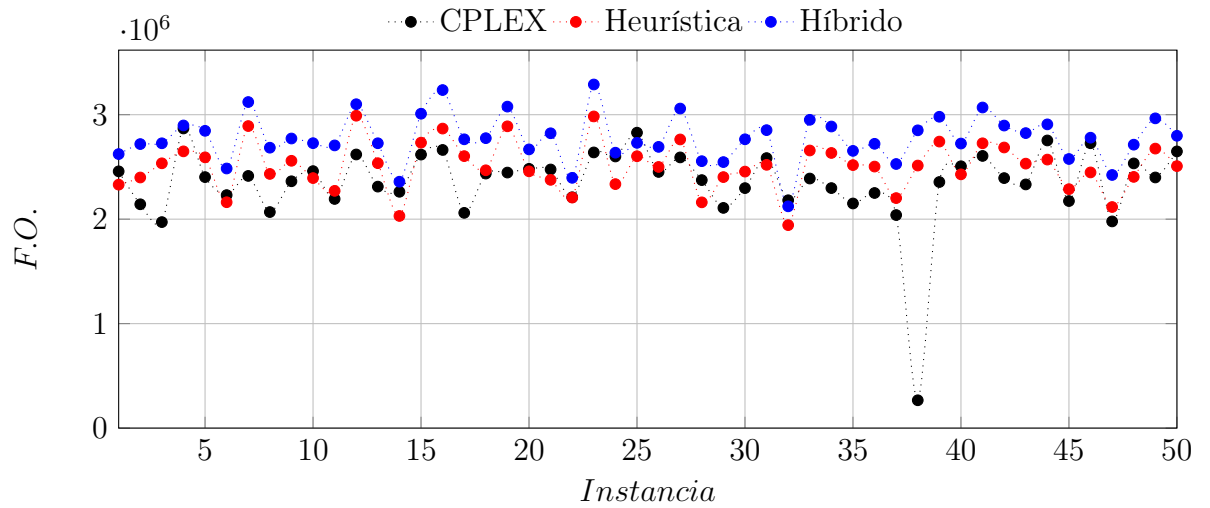


Figura 4.2: Valor de función objetivo para instancias en clase B.

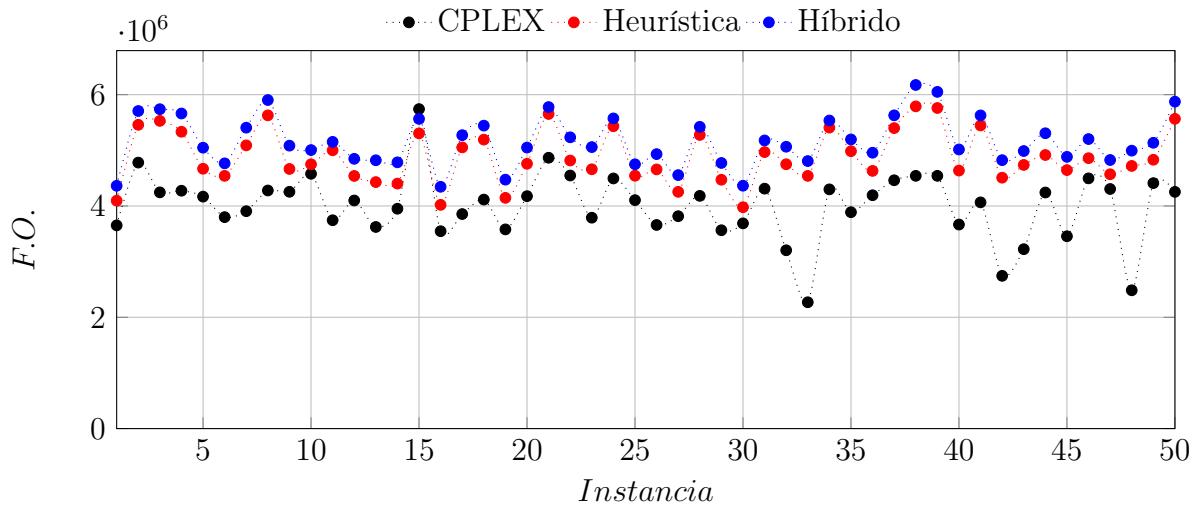


Figura 4.3: Valor de función objetivo para instancias en clase C.

En la Tabla 4.2 se muestra la cantidad de instancias en las que cada método encuentra una función objetivo de mejor calidad. Como se mencionó anteriormente, CPLEX tiene un dominio claro sobre los otros dos métodos en las instancias de clase A (encontró 50/50 mejores soluciones), sin embargo, para las clases de instancias restantes el método dominante es el híbrido obteniendo 48/50 y 49/50 mejores soluciones en clases B y C, respectivamente.

Tabla 4.2: Número de veces en las que el método reportó mejores valores de la función objetivo.

	Clase A	Clase B	Clase C
CPLEX	50/50	2/50	1/50
Heurística	0/50	0/50	0/50
Híbrido	0/50	48/50	49/50

La Tabla 4.3 contiene resultados comparativos (porcentajes de Gap entre soluciones). Para obtener la brecha relativa entre CPLEX y el método híbrido propuesto se utiliza la fórmula:

$$Gap(\%) = 100 \times \left[\frac{Z_C - Z_{Hy}}{Z_C} \right] \quad (4.2)$$

donde Z_C y Z_{Hy} representan los valores de la función objetivo reportados por CPLEX y el híbrido, respectivamente (ver Tablas A.1-A.3 en Apéndice A). De manera similar para las columnas restantes se usaron las fórmulas

$$Gap(\%) = 100 \times \left[\frac{Z_{Hu} - Z_{Hy}}{Z_{Hu}} \right] \quad (4.3)$$

y

$$Gap(\%) = 100 \times \left[\frac{CC - Z_{Hy}}{CC} \right] \quad (4.4)$$

respectivamente. Donde Z_{Hu} es el valor de la función objetivo de la mejor solución encontrada por la heurística y CC es la cota superior reportada por CPLEX para cada instancia, esta cota es calculada de la manera siguiente:

$$CC = Z_C \times \left[1 + \frac{GAP(\%)}{100} \right] \quad (4.5)$$

Tabla 4.3: Brecha relativa entre soluciones (Gap (%)).

		C vs H_y	H_u vs H_y	CC vs H_y
Clase A	Promedio	4.21	-14.70	5.09
	Mínimo	1.49	-22.47	2.56
	Máximo	6.64	-6.34	7.03
Clase B	Promedio	-35.37	-10.98	5.53
	Mínimo	-969.97	-19.16	3.30
	Máximo	3.44	-3.69	7.65
Clase C	Promedio	-31.34	-5.90	4.99
	Mínimo	-111.76	-9.71	3.25
	Máximo	3.07	-2.20	7.17

De la Tabla 4.3 se puede concluir que, aunque CPLEX tiene un dominio sobre la calidad de las soluciones reportadas en las instancias de clase A, este desempeño

se ve afectado con el incremento del tamaño de las instancias, es decir, en clases B y C los métodos de solución restantes alcanzan soluciones de mejor calidad.

Al comparar la heurística presentada en el Capítulo 3 con el método alternativo de esta sección se puede concluir que el método híbrido domina contundentemente a la heurística adaptativa tanto en calidad de soluciones como en tiempo computacional requerido. Más aún, el tiempo computacional promedio requerido por el híbrido ronda en el 10 % del tiempo promedio requerido por la heurística.

Dado que CPLEX reporta valores de Gap (%) altos para las instancias en clases B y C, y a que el híbrido domina en calidad de soluciones obtenidas a la heurística, en la Tabla 4.3 se hace la comparación entre el híbrido y la cota reportada por CPLEX. Se puede concluir que en promedio el valor de la función objetivo reportado por el método híbrido se encuentra a menos del 6 % de la cota reportada por CPLEX, y por tanto, en promedio a menos del 6 % del óptimo.

Finalmente, en las soluciones reportadas por los tres métodos de solución, la cantidad de alimento donada a cada uno de los beneficiarios servidos se ajusta bastante bien a la cantidad mínima de alimentos requeridos por beneficiario, reportando una desviación máxima para la heurística, el híbrido y CPLEX de 0.23 %, 3 % y 4.37 %, respectivamente.

CAPÍTULO 5

REFORMULACIÓN DEL M-CCTP

En este capítulo se presenta una reformulación del Problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP por sus siglas en inglés) introducido por Flores-Garza et al. [14]. Este problema está orientado en la pronta respuesta frente a un desastre con el fin de realizar una entrega eficiente de artículos de primera necesidad (agua, alimentos, medicamentos, ropa, productos de higiene personal, etc.) a la población tanto para reducir los riesgos para la salud como para aumentar su probabilidad de supervivencia.

En la Sección 5.1 se presenta una descripción formal del m-CCTP. En las Secciones 5.2-5.3 se presenta la formulación matemática y el modelo de optimización propuestos por Flores-Garza et al. [14]. En la Sección 5.4 se presenta la reformulación, propuesta en esta tesis, basada en restricciones de flujo en redes. En la Sección 5.5 se presentan desigualdades válidas para fortalecer la reformulación de la Sección 5.4. Finalmente, en la Sección 5.6 se presentan y discuten resultados computacionales.

5.1 DESCRIPCIÓN DEL PROBLEMA

El Problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP) consiste en determinar un conjunto de rutas tal que la suma de tiempos de llegada

a cada nodo es minimizada (es decir se busca minimizar la latencia) tomando en cuenta que el tiempo de duración de cada ruta no debe exceder un tiempo límite preestablecido. En este problema se consideran tres conjuntos de nodos, en primer lugar se encuentran los nodos obligatorios que deben ser incluidos en alguna de las rutas, por otro lado están los nodos opcionales los cuales pueden ser incluidos en una ruta en caso de ser necesario y en última instancia están los nodos inalcanzables que, aunque no pueden incluirse en las rutas, deben de ser cubiertos por al menos uno de los nodos opcionales visitados.

5.2 FORMULACIÓN MATEMÁTICA

Sea $G = (V, E)$ un grafo completo no dirigido donde $V = \{0, 1, 2, \dots, n + 1\}$ es el conjunto de nodos, 0 y $n + 1$ son copias del depósito, $V' = V \setminus \{0, n + 1\}$ es el conjunto de nodos que pueden ser visitados y E es el conjunto de arcos. W es el conjunto que contiene a los nodos inalcanzables que deben ser cubiertos. Sea $T \subset V'$ el conjunto de nodos obligatorios, entonces $V' \setminus T$ es el conjunto de nodos opcionales. El problema consiste entonces en encontrar un conjunto de rutas que incluyan a todos los nodos de T y a los nodos necesarios de $V' \setminus T$ de tal modo que todos los nodos en W sean cubiertos y se minimice la latencia (suma de los tiempos de llegada a los nodos visitados).

Parámetros

$|K|$ cantidad de vehículos disponibles.

t_{ij} tiempo de viaje de $i \in V$ a $j \in V$.

L tiempo máximo de duración de una ruta.

$$a_{il} = \begin{cases} 1 & \text{si la localidad } l \in W \text{ puede ser cubierta por la} \\ & \text{localidad } i \in V', \\ 0 & \text{en otro caso.} \end{cases}$$

Variables de decisión

$$x_{ij}^k = \begin{cases} 1 & \text{si el arco } (i,j) \text{ es recorrido por el vehículo } k \in K, \\ 0 & \text{en otro caso.} \end{cases}$$

$$y_i^k = \begin{cases} 1 & \text{si el nodo } i \in V \text{ es visitado por el vehículo } k \in K, \\ 0 & \text{en otro caso.} \end{cases}$$

s_i^k : tiempo de llegada del vehículo $k \in K$ al nodo $i \in V$.

5.3 MODELO DE OPTIMIZACIÓN

$$\text{m-CCTP} \quad \text{Minimizar } Z = \sum_{k \in K} \sum_{i \in V'} s_i^k \tag{5.1}$$

sujeto a:

$$\sum_{\substack{i \in V' \cup \{0\} \\ i \neq j}} x_{ij}^k - \sum_{\substack{i \in V' \cup \{n+1\} \\ i \neq j}} x_{ji}^k = 0 \quad j \in V', k \in K \tag{5.2}$$

$$\sum_{j \in V' \cup \{n+1\}} x_{ij}^k = y_i^k \quad i \in V', k \in K \tag{5.3}$$

$$\sum_{k \in K} y_i^k \leq 1 \quad i \in V' \setminus \tag{5.4}$$

$$\sum_{k \in K} y_i^k = 1 \quad i \in T \tag{5.5}$$

$$\sum_{i \in V' \setminus T} \left(a_{il} \sum_{k \in K} y_i^k \right) \geq 1 \quad l \in W \tag{5.6}$$

$$\sum_{j \in V'} x_{0j}^k \geq x_{ij}^k \quad i \in V', j \in V \setminus \{0, i\}, \tag{5.7}$$

$$s_i^k + t_{ij} - L(1 - x_{ij}^k) \leq s_j^k \quad i \in V \setminus \{n+1\}, \tag{5.8}$$

$$j \in V \setminus \{0, i\}, k \in K$$

$$s_{n+1}^k \leq L \quad k \in K \tag{5.9}$$

$$\sum_{j \in V \setminus \{0\}} x_{0j}^k \leq 1 \quad k \in K \tag{5.10}$$

$$s_0^k = 0 \quad k \in K \quad (5.11)$$

$$s_i^k \geq 0 \quad i \in V, k \in K \quad (5.12)$$

$$x_{ij}^k \in \{0, 1\} \quad k \in K, i \in V' \cup \{0\}, \\ j \in V \setminus \{0, i\} \quad (5.13)$$

$$y_i^k \in \{0, 1\} \quad k \in K, i \in V' \quad (5.14)$$

La función objetivo (5.1) es la suma de tiempos de llegada a cada uno de los nodos visitados (latencia). Las restricciones (5.2) garantizan la conservación de flujo en los nodos visitados y las restricciones (5.3) aseguran la asignación de nodos a vehículos. Restricciones (5.4) aseguran que un nodo opcional es visitado a lo más una vez, mientras que las restricciones (5.5) aseguran que cada nodo obligatorio (T) es visitado exactamente una vez. Las restricciones de cobertura (5.6) aseguran que cada uno de los nodos inalcanzables (W) es cubierto por al menos uno de los nodos opcionales. Restricciones (5.7) aseguran que cada una de las rutas inicia en el depósito. Las restricciones (5.8) aseguran la consistencia en los tiempos de llegada, además de imponer conectividad en cada una de las rutas. Restricciones (5.9) garantizan que el tiempo de duración de cada ruta no excede al tiempo máximo preestablecido L . Restricciones (5.10) garantizan la creación de una única ruta por cada vehículo disponible. Las restricciones restantes (5.11-5.14) imponen condiciones sobre la naturaleza de las variables.

Cabe mencionar que en el trabajo que se ha tomado como base para esta tesis [14] se realizó experimentación computacional en tres variantes. En primer lugar con la formulación matemática m-CCTP, después con una relajación del m-CCTP (Rm-CTP) en la que se relaja la variable binaria y_i^k , y finalmente con una heurística basada en GRASP. Los resultados computacionales obtenidos con el GRASP, así como le mejor solución obtenida por CPLEX, ya sea resolviendo el m-CCTP o el Rm-CCTP, se encuentran en las Tablas B.1-B.4 en el Apéndice B.

5.4 REFORMULACIÓN

Para reformular el m-CCTP como un modelo con restricciones de flujo en redes se requiere como cambio principal la transformación de nodos obligatorios a nodos opcionales. En este trabajo se propone una transformación que genera un nodo ficticio por cada uno de los nodos obligatorios, cada nodo ficticio es marcado como inalcanzable siendo posible cubrirlo solamente por el nodo opcional (antes obligatorio) para el que fue generado.

De la misma manera que en el trabajo de Flores-Garza et al. [14], se considera $G = (V, E)$ como un grafo completo no dirigido donde, $V = \{0, 1, 2, \dots, n + 1\}$ es el conjunto de nodos, siendo 0 y $n + 1$ copias del depósito, $V' = V \setminus \{0, n + 1\}$ es el conjunto de nodos que pueden ser visitados y E es el conjunto de arcos. W_a es el conjunto que contiene a los nodos inalcanzables que deben ser cubiertos. Es importante recalcar que ahora el conjunto W_a , es el conjunto W aumentado, el cual además de contener a los nodos originalmente inalcanzables, contiene también a los nodos ficticios generados. El problema reformulado consiste entonces en encontrar un conjunto de rutas formadas por todos los nodos necesarios de V' de tal modo que todos los nodos en W_a sean cubiertos y se minimice la latencia. La reformulación queda de la manera siguiente.

Parámetros

$|K|$ cantidad de vehículos disponibles.

t_{ij} tiempo de viaje de $i \in V$ a $j \in V \setminus \{i\}$.

L tiempo máximo de duración de una ruta.

$$a_{il} = \begin{cases} 1 & \text{si la localidad } l \in W_a \text{ puede ser cubierta por la} \\ & \text{localidad } i \in V', \\ 0 & \text{en otro caso.} \end{cases}$$

Variables de decisión

$$x_{ij} = \begin{cases} 1 & \text{si el arco (i,j) es recorrido por alguna de las rutas,} \\ 0 & \text{en otro caso.} \end{cases}$$

s_i : tiempo de llegada al nodo $i \in V$.

$$\text{m-CCTP(NF) Minimizar } Z = \sum_{i=1}^n s_i \quad (5.15)$$

suje to a:

$$\sum_{j=1}^n x_{0j} = |K| \quad (5.16)$$

$$\sum_{j=0, j \neq i}^n x_{ji} - \sum_{j=1, j \neq i}^{n+1} x_{ij} = 0 \quad i \in V' \quad (5.17)$$

$$\sum_{i=0, i \neq j}^n x_{ij} \leq 1 \quad j \in V' \quad (5.18)$$

$$\sum_{i=1}^n \left(a_{il} \sum_{j=1, j \neq i}^{n+1} x_{ij} \right) \geq 1 \quad l \in W_a \quad (5.19)$$

$$s_i + t_{ij} - L(1 - x_{ij}) \leq s_j \quad i \in V \setminus \{n+1\}, \\ j \in V \setminus \{0, i\} \quad (5.20)$$

$$s_i + t_{i(n+1)} \leq L \quad i \in V' \quad (5.21)$$

$$s_{n+1} \leq L \quad (5.22)$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (5.23)$$

$$s_i \geq 0 \quad i \in V \quad (5.24)$$

La función objetivo (5.16) es la suma de los tiempos de llegada a los nodos visitados (latencia). Las restricciones (5.16) aseguran que del depósito salgan $|K|$ unidades de flujo, es decir, una unidad de flujo por cada vehículo disponible. Restricciones (5.17) garantizan la conservación de flujo en cada nodo opcional. Las restricciones (5.18) limitan la entrada a cada nodo opcional en a lo más una unidad de flujo. Las restricciones (5.19) por su parte aseguran que cada uno de los nodos inalcanzables sea cubierto por al menos uno de los nodos opcionales. Las restricciones (5.20) aseguran la consistencia en los tiempos de llegada, además de imponer

conectividad en cada una de las rutas. Las restricciones (5.21-5.22) imponen condiciones sobre el máximo tiempo de llegada admisible a cada nodo. Finalmente, las restricciones (5.23-5.24) imponen condiciones sobre la naturaleza de las variables.

5.5 DESIGUALDADES VÁLIDAS

Con el fin de fortalecer la reformulación propuesta en la sección anterior, se propone una serie de desigualdades válidas. Dado que la restricción (5.18) asegura que a cada nodo entra a lo más una unidad de flujo y que las restricciones (5.17) garantizan la conservación de flujo, las restricciones (5.25) son desigualdades válidas que proporcionan una cota superior a la cantidad de flujo que puede salir de cada nodo.

$$\sum_{i=1, i \neq j}^{n+1} x_{ji} \leq 1 \quad j \in V' \quad (5.25)$$

Sea la ruta presentada en la Figura 5.1, que en este caso consta de tres nodos a visitar, una ruta a seguir por alguno de los vehículos de la instancia, cada uno de los arcos de la ruta tiene una etiqueta que representa la longitud (duración) del mismo. La Tabla 5.1 muestra el tiempo de llegada a cada uno de los nodos de la ruta, además de la duración total y la latencia de la misma. Nótese que la duración total de la ruta coincide con el tiempo de llegada al último nodo visitado, de tal modo que la duración total de la ruta representa una cota inferior para la latencia. Por otro lado, a excepción del primer nodo visitado, cada uno de los nodos de la ruta tiene un tiempo de llegada acumulado (dado por los tiempos de llegada a los nodos anteriores), este tiempo de llegada es, en el mejor de los casos, el tiempo mínimo sobre los arcos de la red. Basados en el hecho de que se tienen $|K|$ rutas, entonces se tienen exactamente $|K|$ nodos iniciales visitados en el problema abordado.

Sin pérdida de generalidad los resultados presentados para la Figura 5.1 se pueden extender, de tal modo que las restricciones (5.26-5.27) proveen cotas inferiores

para la función objetivo de latencia.

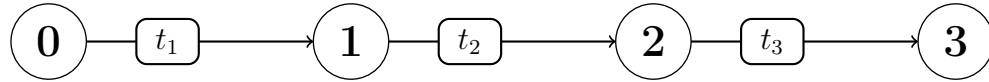


Figura 5.1: Ruta a seguir por un vehículo.

Nodo	Tiempo de llegada
1	t_1
2	$t_1 + t_2$
3	$t_1 + t_2 + t_3$
Duración	$t_1 + t_2 + t_3$
Latencia	$(3)t_1 + (2)t_2 + (1)t_3$

Tabla 5.1: Tiempos de llegada a cada nodo de la Figura 5.1.

La restricción (5.26) provee el tiempo total de ruta como cota inferior para latencia.

$$\sum_{i=1}^n s_i \geq \sum_{i=0}^n \sum_{j=1, j \neq i}^n t_{ij} x_{ij} \quad (5.26)$$

La restricción (5.27) además de utilizar la distancia total recorrida, agrega una componente que refleja el mínimo tiempo de llegada acumulado antes de arribar a algún nodo visitado.

$$\sum_{i=1}^n s_i \geq \sum_{i=0}^n \sum_{j=1, j \neq i}^n t_{ij} x_{ij} + \min_{(i,j)} \{t_{ij}\} \left(\sum_{i=0}^n \sum_{j=1, j \neq i}^n x_{ij} - |K| \right) \quad (5.27)$$

Por su parte las restricciones (5.28) explotan la condición de que cualquier arco del grafo puede ser utilizado en a lo más un sentido, siendo entonces incompatibles las variables correspondientes a ambos sentidos en un mismo arco.

$$x_{ij} + x_{ji} \leq 1 \quad i, j \in V, i \neq j \quad (5.28)$$

Dado que cada una de las rutas no debe exceder el tiempo máximo de duración L , una cota superior para la distancia recorrida está dada por (5.29).

$$\sum_{i=0}^n \sum_{j=1, j \neq i}^{n+1} t_{ij} x_{ij} \leq L \times |K| \quad (5.29)$$

Las desigualdades válidas antes descritas aportan cotas para salida de flujo de un nodo, cotas inferiores para la función objetivo de interés (en este caso latencia), cotas para la distancia total recorrida y restricciones para conjuntos de variables incompatibles. A continuación se presenta una restricción que refleja una cota superior sobre el número de arcos intermedios que pueden ser utilizados en una solución factible.

$$\sum_{i=1}^{n-1} \sum_{j=2, j \neq i}^n x_{ij} \leq n - |K| \quad (5.30)$$

Estas desigualdades propuestas se introdujeron a la reformulación propuesta en la Sección 5.4. Los resultados computacionales se muestran a continuación.

5.6 RESULTADOS COMPUTACIONALES

Con el fin de evaluar la reformulación propuesta para el m-CCTP, así como el impacto de las desigualdades válidas descritas en la sección anterior, se implementaron los modelos en lenguaje C++ y se resolvieron las instancias mediante el optimizador CPLEX en su versión 12.6 en una CPU Intel Xeon(R) a 2.10 GHz E5-2620 v2 bajo el sistema operativo Ubuntu 14.04 LTS. Los criterios de paro para cada una de las instancias fueron dos horas de tiempo de cómputo (7200 s) y/o un Gap de 0.01 %. Las instancias de prueba fueron adaptadas por Flores-Garza et al. [14] del trabajo publicado por Ha et al. [20] y están seccionadas en 4 clases, 32 instancias para cada una de las clases A y B, y 16 instancias para cada una de las clases C y D.

Como se mencionó anteriormente, en el trabajo de Flores-Garza et al. [14] se realizó experimentación computacional con tres variantes. En primer lugar con la formulación matemática m-CCTP, después con una relajación del m-CCTP (Rm-CCTP), en la que se relaja la variable binaria y_i^k , y finalmente con una heurística basada en GRASP. Los resultados computacionales obtenidos con el GRASP, así como la mejor solución obtenida por CPLEX, ya sea resolviendo el m-CCTP o el Rm-CCTP, se encuentran en las Tablas B.1-B.4 en el Apéndice B.

Por simplicidad para referirse a la mejor solución encontrada por las formulaciones propuestas por Flores-Garza et al. [14] se utilizará m-CCTP, para la reformulación propuesta en esta tesis se utilizará m-CCTP(NF) y finalmente para la reformulación propuesta fortalecida con las desigualdades válidas (5.25-5.30) se utilizará m-CCTP(NF+DV).

La Tabla 5.2 muestra la cantidad de instancias en las que CPLEX encuentra solución entera para las formulaciones mencionadas.

Tabla 5.2: Soluciones enteras encontradas para cada clase de instancias.

Clase	m-CCTP	m-CCTP(NF)	m-CCTP(NF+DV)
A	13/32	26/32	25/32
B	14/32	22/32	26/32
C	9/16	16/16	16/16
D	10/16	16/16	16/16

Nótese que tanto la formulación m-CCTP(NF) como m-CCTP(NF+DV) superan contundentemente a la formulación m-CCTP en cuanto a soluciones enteras encontradas por el optimizador CPLEX, además, si tomamos en cuenta ambas formulaciones propuestas en esta tesis el dominio se incrementa al llegar al resultado de que en total para la clase A se encuentran 28/32 soluciones y para las clases B, C y D se encuentran 26/32, 16/16 y 16/16 soluciones enteras, respectivamente.

Por otro lado, además de que con las formulaciones propuestas se encuentra un mayor número de soluciones enteras, el valor del Gap (%) reportado por CPLEX es, en promedio, más bajo que el de las formulaciones propuestas por Flores-Garza et al. [14] (ver Tabla 5.3). Cabe mencionar que las desigualdades válidas (5.25-5.30) se reflejan en una disminución de al menos 20% de Gap con respecto a la reformulación m-CCTP(NF), lo cuál indica que dichas desigualdades mejoran el desempeño de CPLEX. Más aún, con la formulación m-CCTP(NF+DV) se llega al óptimo en tres de las instancias de prueba (ver Tablas B.5-B.9 en el Apéndice B).

Tabla 5.3: Gap (%) promedio reportado por CPLEX para cada clase de instancias.

Clase	m-CCTP	m-CCTP(NF)	m-CCTP(NF+DV)
A	88.53	84.10	55.71
B	90.53	86.00	63.23
C	78.94	76.53	44.71
D	89.35	84.02	64.10

Dado que las soluciones reportadas en el trabajo de referencia se obtienen en su mayoría por una heurística basada en un GRASP, en la Tabla 5.4 se muestra el Gap (%) que hay entre los valores de la función objetivo de las soluciones encontradas, si $M1$ y $M2$ son los métodos a comparar, el Gap (%) de las soluciones encontradas con los métodos $M1$ vs $M2$ se calcula de la manera siguiente:

$$Gap(\%) = 100 \times \left[\frac{FO_{M1} - FO_{M2}}{FO_{M1}} \right] \quad (5.31)$$

Tabla 5.4: Gap (%) promedio entre métodos por clase de instancia.

Clase	GRASP vs m-CCTP(NF+DV)	GRASP vs MSE
A	-12.15	-17.64
B	-17.10	-16.56
C	-2.77	-1.63
D	-10.57	-8.85

Nótese que debido a la manera de calcular el Gap (%) entre soluciones y a que el problema tiene una función objetivo de minimización, un valor negativo indica que el GRASP alcanza un valor de función objetivo de mejor calidad. Según los datos mostrados en la Tabla 5.4, es evidente que el GRASP le gana a la reformulación propuesta aún cuando está fortalecida con desigualdades válidas m-CCTP(NF+DV). Más aún, el GRASP resulta vencedor incluso si lo comparamos contra la mejor solución encontrada (MSE) por las formulaciones propuestas en esta tesis. A pesar que en promedio el GRASP siempre resulta victorioso vale la pena hacer comparaciones puntuales, las cuales se muestran a continuación.

Aunque en promedio las formulaciones propuestas tienen un desempeño por debajo del GRASP propuesto por Flores-Garza et al. [14], el m-CCTP(NF+DV) encuentra soluciones tan buenas como el GRASP, incluso en algunas instancias encuentra soluciones de mejor calidad que las reportadas en la literatura. En la Tabla 5.5 se muestra el número de instancias en las que se encuentran con la formulación m-CCTP(NF+DV) soluciones al menos tan buenas como las del GRASP y soluciones en las que incluso supera a las soluciones reportadas en la literatura (GRASP o m-CCTP).

Tabla 5.5: Comparación puntual entre soluciones.

	No peor a GRASP		Supera a GRASP	
Clase	m-CCTP(NF+DV)	MSE	m-CCTP(NF+DV)	MSE
A	11	14	5	8
B	9	9	5	5
C	10	10	2	3
D	4	6	1	2

Véase que si tomamos en cuenta la mejor solución encontrada por las formulaciones propuestas en esta tesis, se tiene que en 39/96 instancias se alcanza una solución con un valor de función objetivo no peor al encontrado por el GRASP. Sin embargo, esto no es lo más rescatable, lo más notable es que en 18/96 instancias se encuentra una solución con un valor de función objetivo mejor que las reportadas en literatura lo que hace relevante la introducción de estas formulaciones.

CAPÍTULO 6

CONCLUSIONES Y TRABAJO FUTURO

En esta tesis se estudiaron dos temas relacionados con la logística humanitaria, por un lado se introdujo el problema de la asignación de recursos en un banco de alimentos que no había sido abordado en la literatura. Por otro lado, se presentó una reformulación para el problema de recorridos de cobertura con tiempos acumulativos.

Uno de los objetivos de este trabajo fue presentar una alternativa para la distribución eficiente de recursos en los bancos de alimentos, todo esto tomando en cuenta beneficiarios con diferentes características y necesidades, inventario existente y presupuesto para comprar algunos productos a fin de garantizar la seguridad alimenticia y la nutrición balanceada a las personas con inseguridad alimentaria. Se propuso una formulación lineal entera mixta, una heurística adaptativa y un método de solución alternativo que incluye componentes heurísticos y componentes de solución exacta. Se evaluó el desempeño de cada uno de ellos sobre un conjunto grande de instancias, generadas aleatoriamente con base en la situación real de un banco de alimentos local. Los resultados computacionales obtenidos revelan que la formulación propuesta se puede usar para resolver instancias de tamaño mediano (con 200 diferentes productos y 100 beneficiarios). Sin embargo, en instancias grandes, tanto la heurística adaptativa como el método híbrido superan a CPLEX con un promedio de hasta 35.37 % de Gap utilizando un tiempo computacional significativamente más corto que el requerido por CPLEX.

Adicionalmente, se observó que dadas las similitudes que el problema de estudio tiene con el problema de la mochila (Knapsack Problem), los beneficiarios servidos que se reportan en la solución final encontrada por los métodos de solución propuestos son los beneficiarios que tienen un alto valor del radio vulnerabilidad/demanda (W_n/D_n). Por lo tanto, en la práctica, una buena solución podría obtenerse simplemente tratando de cubrir la demanda de los beneficiarios que tienen los valores más altos de la relación mencionada.

Los resultados de este trabajo, correspondientes al problema de asignación de recursos en un banco de alimentos, se han enviado a evaluación para su posible publicación en la revista *Computers & Operations Research*.

El alcance futuro de esta investigación, con respecto al FBRAP, es hacer una extensión para diferenciar y priorizar los productos de acuerdo con su grado de perecedencia. Esto se debe a que la mayoría de los productos que llegan a un banco de alimentos provienen de cadenas comerciales y en su mayoría tienen un período de tiempo muy corto en el que son aptos para el consumo.

En esta tesis se propuso también una reformulación del problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP), la cual está basada en restricciones de flujo en redes. Se fortalece con desigualdades válidas y se evalúa el alcance de los modelos propuestos a través de la experimentación computacional con un conjunto grande de instancias. Los resultados obtenidos muestran que con las formulaciones propuestas se encuentran soluciones cuyo valor de función objetivo supera, en calidad, a algunas de las mejores soluciones reportadas en la literatura. Además, con este trabajo se ha podido garantizar la optimalidad de algunos de los resultados obtenidos.

Como futura extensión de la parte correspondiente al problema de recorridos de cobertura con tiempos acumulativos se encuentra el diseño e implementación de un algoritmo de generación de columnas.

APÉNDICE A

TABLAS DE RESULTADOS DEL FBRAP

En las Tablas A.1-A.3 se presentan los resultados computacionales para el Problema de Asignación de Recursos en Bancos de Alimentos (FBRAP), para las instancias de clase A, B y C, respectivamente.

En la primer columna se encuentra el nombre de la instancias, dado que todas las instancias están formadas por el mismo número de productos (200) se utiliza para identificarlas el formato: “*beneficiarios_identificador*”, por ejemplo la instancia “300_4” está formada por 200 tipos productos, 300 beneficiarios y el 4 se utiliza como identificador. La segunda y tercer columna de las tablas muestran el valor de la función objetivo y el Gap (%) reportado por el optimizador CPLEX en cada una de las instancias. Las columnas restantes contienen el valor de la función objetivo de la mejor solución encontrada, así como el tiempo computacional utilizado por cada uno de los métodos de solución propuestos. Cabe mencionar que el tiempo computacional en el que CPLEX encontró cada de las soluciones fue de 7200 s (2 horas) para cada instancia de prueba.

Tabla A.1: Resultados computacionales para FBRAP para instancias de clase A.

Instancia	CPLEX		Heurística		Híbrido	
	Z_C	Gap (%)	Z_{He}	Tiempo (s)	Z_{Hy}	Tiempo (s)
100_1	1169900	0.71	961800	191.63	1113800	10.31
100_2	1040800	1.25	886700	83.47	999700	7.47
100_3	986300	1.29	812500	162.31	946300	16.02
100_4	1214900	1.68	971800	99.55	1154400	10.10
100_5	919600	0.78	758300	107.20	885900	22.99
100_6	1055000	1.35	909700	81.55	1017300	6.79
100_7	1000500	0.27	838200	125.27	961400	7.09
100_8	1183000	1.40	972000	134.50	1121200	7.22
100_9	1141000	1.08	933200	72.07	1086300	9.76
100_10	978000	1.09	864100	88.38	963400	20.82
100_11	930100	1.18	749300	226.06	884000	14.86
100_12	1128000	0.87	904900	73.41	1058700	6.70
100_13	1078300	1.20	899500	74.57	1034200	12.25
100_14	957100	1.53	778000	99.18	924800	18.94
100_15	1158100	0.93	1002700	109.27	1112400	10.99
100_16	867000	0.93	714200	145.55	819500	9.53
100_17	1065500	0.43	821000	102.41	1005500	7.52
100_18	1113600	1.35	903000	82.32	1055700	13.89
100_19	1094800	1.31	920900	108.15	1048600	7.27
100_20	1185200	0.30	981700	71.00	1133900	6.17
100_21	1005900	0.30	831500	106.77	960200	8.18
100_22	1161900	0.93	960500	80.47	1107100	9.86
100_23	1123900	0.57	971000	102.88	1080700	10.81
100_24	1152400	1.11	989200	83.85	1101500	14.40
100_25	1041300	0.52	828100	182.19	988300	8.13

Continúa en la siguiente página

Tabla A.1, continuación

Instancia	CPLEX		Heurística		Híbrido	
	Z_C	Gap (%)	Z_{He}	Tiempo (s)	Z_{Hy}	Tiempo (s)
100_26	935100	1.12	784200	133.55	901000	15.15
100_27	1017000	0.29	895400	100.66	983500	7.76
100_28	1030300	0.05	863700	126.61	978900	7.77
100_29	1009300	1.80	861100	120.92	992300	14.72
100_30	1179800	1.63	969100	79.06	1130900	19.40
100_31	1087400	0.26	960200	112.92	1062000	8.96
100_32	1093600	1.21	893100	112.35	1047200	12.86
100_33	999800	0.52	817900	106.86	934300	11.35
100_34	974400	0.50	877400	136.53	933000	8.64
100_35	1021600	0.56	849900	109.80	970600	11.09
100_36	964900	1.92	802300	142.18	926000	16.31
100_37	1129100	1.34	995300	60.71	1098900	13.07
100_38	1097200	0.66	932300	91.39	1056100	7.36
100_39	1105700	2.40	979800	50.44	1087300	9.45
100_40	920300	0.40	717700	130.91	859200	9.55
100_41	1098700	0.45	930200	69.12	1057500	7.90
100_42	952800	0.32	797000	46.07	908300	9.43
100_43	1146600	0.92	962300	108.76	1090600	11.26
100_44	1020100	0.40	837900	108.65	977200	20.94
100_45	920100	0.10	776400	60.06	887700	16.53
100_46	1014000	1.77	809400	70.31	971200	8.54
100_47	1033300	0.45	892400	48.14	992900	7.53
100_48	928900	0.62	774300	91.84	884600	11.47
100_49	1092000	1.53	923100	61.36	1054800	11.61
100_50	1118500	1.24	942800	155.14	1077400	8.91

Tabla A.2: Resultados computacionales para FBRAP para instancias de clase B.

Instancia	CPLEX		Heurística		Híbrido	
	Z_C	Gap (%)	Z_{He}	Tiempo (s)	Z_{Hy}	Tiempo (s)
300_1	2457700	14.47	2330300	972.329	2623000	71.0881
300_2	2142500	34.69	2398800	360.747	2718800	58.8822
300_3	1971900	44.09	2534700	1120.94	2726300	89.1186
300_4	2868200	7.18	2649000	979.351	2896900	59.1287
300_5	2402600	27.64	2591800	788.688	2845700	50.4843
300_6	2230200	17.36	2162100	748.02	2485000	64.6073
300_7	2413900	37.12	2891200	532.148	3121200	70.6754
300_8	2068100	34.85	2433000	826.282	2684100	93.1895
300_9	2362700	24.19	2558800	724.597	2772000	70.5692
300_10	2459900	17.07	2393200	467.828	2727400	102.267
300_11	2193400	32.14	2270900	497.62	2706000	81.8629
300_12	2619200	22.72	2990100	318.873	3100300	47.8165
300_13	2311500	23.97	2535800	1241.38	2726600	96.8405
300_14	2262000	10.46	2030700	957.897	2358500	59.2278
300_15	2618100	23.66	2732300	459.53	3009100	59.6059
300_16	2662600	28.81	2867000	904.851	3235900	54.9594
300_17	2060300	38.75	2603300	429.577	2764400	54.5076
300_18	2436400	20.91	2467900	264.361	2775300	50.44
300_19	2446000	32.95	2888700	1245.54	3077300	87.2264
300_20	2481700	15.71	2458000	523.147	2667800	72.0808
300_21	2475500	22.27	2376300	616.651	2822200	69.6112
300_22	2209600	13.22	2206800	1383.72	2395600	68.5102
300_23	2638800	32.68	2983400	1132.42	3289100	43.1372
300_24	2600100	7.68	2335500	848.479	2636300	51.4549
300_25	2827900	1.35	2602400	702.811	2730500	46.6334

Continúa en la siguiente página

Tabla A.2, continuación

Instancia	CPLEX		Heurística		Híbrido	
	Z_C	Gap (%)	Z_{He}	Tiempo (s)	Z_{Hy}	Tiempo (s)
300_26	2451900	14.16	2499800	465.954	2691900	58.4698
300_27	2591100	25.23	2763000	1100.77	3058700	101.025
300_28	2373900	15.88	2161000	1039.1	2555100	62.1092
300_29	2107800	25.36	2402700	490.565	2546400	25.7634
300_30	2297200	27.24	2455300	570.431	2764600	69.207
300_31	2584600	19.43	2520300	805.881	2852100	69.1947
300_32	2181100	1.50	1942900	1037.64	2122800	49.7764
300_33	2388900	29.80	2657300	396.86	2951100	72.9035
300_34	2297500	34.98	2633700	639.902	2886900	81.686
300_35	2150100	28.63	2517500	488.524	2653600	33.6432
300_36	2250500	26.79	2503400	352.885	2721300	89.4127
300_37	2039400	33.73	2201700	1129.55	2528000	70.6167
300_38	266400	1045.81	2513200	936.585	2850400	63.8969
300_39	2355600	31.56	2742500	512.262	2980300	52.0413
300_40	2506500	15.36	2429000	524.491	2723800	81.6598
300_41	2605300	26.48	2726300	374.277	3068900	74.7928
300_42	2393100	26.82	2686100	694.798	2896000	64.6893
300_43	2332400	28.22	2531500	633.896	2823800	76.2637
300_44	2753400	13.21	2570300	726.181	2907400	103.598
300_45	2173400	23.32	2287200	1426.19	2575900	54.9449
300_46	2726400	8.56	2447400	292.397	2779800	69.4507
300_47	1978400	32.61	2116200	506.671	2423000	93.9402
300_48	2533700	12.52	2404600	1005.99	2713000	38.9796
300_49	2399200	30.19	2675400	1069.99	2965000	58.564
300_50	2648400	12.88	2507700	1367.94	2798200	67.7691

Tabla A.3: Resultados computacionales para FBRAP para instancias de clase C.

Instancia	CPLEX		Heurística		Híbrido	
	Z_C	Gap (%)	Z_{He}	Tiempo (s)	Z_{Hy}	Tiempo (s)
500_1	3651400	24.75	4095000	1071.94	4362700	75.0067
500_2	4779500	24.29	5458500	1108.34	5706400	68.9911
500_3	4244800	43.07	5528900	2815.41	5738900	51.4032
500_4	4274000	40.17	5333800	2040.42	5661400	154.806
500_5	4167900	28.37	4668800	849.175	5047100	63.5911
500_6	3799400	31.22	4541300	3263.68	4765900	111.462
500_7	3906100	46.80	5088900	1479.58	5406900	152.183
500_8	4278200	44.74	5628500	840.611	5903600	80.1492
500_9	4254900	26.38	4665200	677.572	5083700	112.623
500_10	4577900	15.20	4749000	781.904	5004700	125.049
500_11	3743700	44.37	5000100	1202.32	5151800	71.4582
500_12	4098900	27.39	4539400	782.957	4847000	174.28
500_13	3622700	40.45	4429000	1635.46	4821500	131.6
500_14	3950000	27.10	4402000	832.052	4784100	105.68
500_15	5740800	1.51	5304700	1143.89	5564700	82.1875
500_16	3546400	31.36	4020300	1312.38	4344600	135.204
500_17	3854600	45.26	5052800	710.228	5272100	131.955
500_18	4114800	39.22	5192300	1155.52	5443600	92.5981
500_19	3579200	32.26	4146100	1165.7	4472800	191.838
500_20	4176000	27.07	4758500	1290.48	5049700	116.515
500_21	4866500	23.90	5652700	1060.75	5777300	30.6733
500_22	4547300	22.17	4817200	1279.77	5233800	96.8337
500_23	3789300	40.71	4659600	553.364	5057900	52.9034
500_24	4494500	30.92	5432500	712.38	5573500	68.4348
500_25	4105000	20.49	4543400	1018.52	4749100	134.181

Continúa en la siguiente página

Tabla A.3, continuación

Instancia	CPLEX		Heurística		Híbrido	
	Z_C	Gap (%)	Z_{He}	Tiempo (s)	Z_{Hy}	Tiempo (s)
500_26	3658700	40.77	4657000	2474.84	4931700	137.287
500_27	3816300	25.83	4253900	1085.15	4554800	111.633
500_28	4181800	36.13	5275900	612.198	5422000	65.3472
500_29	3564000	41.03	4471500	1084.02	4772600	186.016
500_30	3687400	24.98	3977500	618.406	4363900	186.788
500_31	4311200	26.72	4967800	924.544	5177800	208.568
500_32	3204000	65.63	4749700	1099.31	5064500	76.1231
500_33	2270000	122.50	4541000	1777.97	4806900	107.158
500_34	4298500	33.83	5407800	1055.67	5535800	74.5424
500_35	3888100	40.43	4984300	1186.41	5195500	51.4473
500_36	4192700	24.36	4630200	1546.09	4956500	92.0773
500_37	4461900	33.14	5399600	1693.75	5628900	70.5726
500_38	4541400	43.08	5789600	2079.3	6174200	119.426
500_39	4541800	41.13	5760800	843.106	6050100	155.064
500_40	3666300	46.47	4636100	599.3	5015200	100.339
500_41	4064800	43.67	5446700	801.416	5627700	30.8875
500_42	2744700	85.23	4508300	1554.55	4823400	69.6075
500_43	3223400	61.89	4737700	1421.66	4987200	40.8119
500_44	4242000	32.85	4915800	1075.14	5306800	89.7626
500_45	3455900	46.02	4645000	976.339	4882600	31.0942
500_46	4496400	22.67	4858900	1123.21	5202400	87.5931
500_47	4305000	16.82	4569900	1164	4826000	97.3493
500_48	2484800	110.82	4719500	1375.9	4993800	58.0961
500_49	4410300	21.06	4831000	1229.07	5135400	149.889
500_50	4253600	45.69	5565700	1069.52	5874000	164.214

APÉNDICE B

TABLAS DE RESULTADOS DEL M-CCTP

En este Apéndice se muestran los resultados computacionales para el Problema de Recorridos de Cobertura con Tiempos Acumulativos (m-CCTP)

B.1 PUNTO DE REFERENCIA PARA EL M-CCTP

En las Tablas B.1-B.4 se presentan los resultados computacionales publicados por Flores-Garza et al. [14] m-CCTP, para las instancias de clase A, B, C y D, respectivamente.

En la primer columna se encuentra el nombre de la instancia, el cual está formado por la clase a la que pertenece, el número total de nodos; dicho número se divide a su vez en tres partes, la primera parte refleja la cantidad de nodos que son obligatorios, la segunda contiene el número de nodos opcionales y finalmente el número de nodos inalcanzables, además de un dígito final que no es relevante para nuestro estudio. Por ejemplo la instancia “kroA100-1-24-75-1” pertenece a la clase A, contiene un total de 100 nodos de los cuales uno es obligatorio, 24 opcionales y 75 inalcanzables, dichas instancias fueron adaptadas por Flores-Garza et al. [14] del trabajo publicado por Ha et al. [20]. La segunda columna contiene la cantidad de vehículos disponibles en la instancia. La tercera columna contiene el valor de

la función objetivo de la mejor solución encontrada por su GRASP y la cuarta el tiempo que tardó en encontrarla. Las dos columnas restantes contienen el valor de la función objetivo y el Gap (%), respectivamente, reportados por CPLEX para la mejor solución encontrada por sus formulaciones usando un criterio de paro de dos horas.

Tabla B.1: Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase A.

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroA100-1-24-75-1	2	12240	77.55	13332	88.59
kroA100-1-24-75-2	2	12240	82.52	12240	84.72
kroA100-1-24-75-3	2	12240	82.52	12240	81.9
kroA100-1-24-75-4	1	20141	87.65	20141	86.74
kroA100-1-49-50-1	3	13777	198.82	—	—
kroA100-1-49-50-2	2	20921	216.94	37144	98.75
kroA100-1-49-50-3	2	20443	216.09	31008	97.93
kroA100-1-49-50-4	1	34517	233.24	55169	98.4
kroA200-1-99-100-1	3	18239	773.30	—	—
kroA200-1-99-100-2	2	21319	785.08	—	—
kroA200-1-99-100-3	2	21632	837.06	—	—
kroA200-1-99-100-4	2	21081	841.40	—	—
kroA100-10-40-50-1	5	28603	165.02	—	—
kroA100-10-40-50-2	4	30096	169.14	—	—
kroA100-10-40-50-3	3	34483	188.79	—	—
kroA100-10-40-50-4	3	35110	186.12	—	—
kroA100-5-20-75-1	2	14899	71.77	14899	88.56
kroA100-5-20-75-2	2	14899	68.52	14899	78.32
kroA100-5-20-75-3	2	14899	66.44	14899	77.41
kroA100-5-20-75-4	1	26571	60.83	26571	73.67

Continúa en la siguiente página

Tabla B.1, continuación

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroA200-1-49-150-1	2	20112	328.46	—	—
kroA200-1-49-150-2	2	21458	328.36	—	—
kroA200-1-49-150-3	2	20803	330.52	25480	98.1
kroA200-1-49-150-4	1	35083	359.26	37962	97.82
kroA200-10-40-150-1	4	27437	231.43	—	—
kroA200-10-40-150-2	3	30663	231.49	—	—
kroA200-10-40-150-3	3	30737	227.56	—	—
kroA200-10-40-150-4	2	42013	230.73	—	—
kroA200-20-80-100-1	7	44818	626.48	—	—
kroA200-20-80-100-2	6	45791	635.77	—	—
kroA200-20-80-100-3	5	49616	735.79	—	—
kroA200-20-80-100-4	4	56679	652.74	—	—

Tabla B.2: Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase B.

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroB100-1-24-75-1	2	8836	64.71	8836	81.22
kroB100-1-24-75-2	2	8836	78.06	8836	83.59
kroB100-1-24-75-3	1	12133	94.73	12133	82.77
kroB100-1-24-75-4	1	12133	84.96	12133	80.37
kroB100-1-49-50-1	2	16095	205.58	—	—
kroB100-1-49-50-2	2	16350	221.27	—	—
kroB100-1-49-50-3	2	17219	220.65	22734	98.15
kroB100-1-49-50-4	2	16919	226.03	20571	97.75
kroB200-1-99-100-1	2	55916	864.08	—	—
kroB200-1-99-100-2	2	47850	853.50	—	—

Continúa en la siguiente página

Tabla B.2, continuación

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroB200-1-99-100-3	2	44032	958.04	—	—
kroB200-1-99-100-4	1	90200	997.44	—	—
kroB100-10-40-50-1	4	23872	196.38	—	—
kroB100-10-40-50-2	3	26672	175.75	—	—
kroB100-10-40-50-3	3	26822	173.41	—	—
kroB100-10-40-50-4	2	31544	168.96	71391	98.58
kroB100-5-20-75-1	2	16157	66.20	16238	92.37
kroB100-5-20-75-2	2	16341	75.34	16238	88.11
kroB100-5-20-75-3	2	16157	76.68	15737	85.91
kroB100-5-20-75-4	1	27910	65.37	29409	85.75
kroB200-1-49-150-1	3	15928	295.36	—	—
kroB200-1-49-150-2	2	18860	348.91	—	—
kroB200-1-49-150-3	2	19266	317.10	33447	98.22
kroB200-1-49-150-4	2	19040	308.09	24997	98.44
kroB200-10-40-150-1	2	32407	254.88	—	—
kroB200-10-40-150-2	2	34806	233.92	—	—
kroB200-10-40-150-3	2	34182	226.34	—	—
kroB200-10-40-150-4	1	68245	272.21	105038	96.29
kroB200-20-80-100-1	9	67466	682.52	—	—
kroB200-20-80-100-2	7	66421	700.02	—	—
kroB200-20-80-100-3	6	69781	668.24	—	—
kroB200-20-80-100-4	5	71465	662.52	—	—

Tabla B.3: Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase C.

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroC100-1-24-75-1	1	9529	76.24	9529	75.77
kroC100-1-24-75-2	1	9529	70.91	9529	74.08
kroC100-1-24-75-3	1	9529	72.75	9529	73.23
kroC100-1-24-75-4	1	9529	79.27	9529	74.22
kroC100-1-49-50-1	3	15508	184.71	—	—
kroC100-1-49-50-2	2	18549	188.49	—	—
kroC100-1-49-50-3	2	18764	185.98	—	—
kroC100-1-49-50-4	2	19007	191.06	33218	98.36
kroC100-10-40-50-1	4	29721	175.12	—	—
kroC100-10-40-50-2	4	29660	170.67	—	—
kroC100-10-40-50-3	3	33026	186.43	—	—
kroC100-10-40-50-4	2	45564	191.02	—	—
kroC100-5-20-75-1	2	14311	65.55	14311	81.6
kroC100-5-20-75-2	2	14311	62.15	14311	81.77
kroC100-5-20-75-3	2	14311	67.14	14311	82.17
kroC100-5-20-75-4	1	21794	65.48	21939	69.29

Tabla B.4: Resultados computacionales publicados por Flores-Garza et al. [14] para instancias en clase D.

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroD100-1-24-75-1	2	9661	91.47	9878	79.32
kroD100-1-24-75-2	2	9661	85.59	9661	81.57
kroD100-1-24-75-3	1	16556	86.31	16556	87.93
kroD100-1-24-75-4	1	16556	89.67	16556	86.56
kroD100-1-49-50-1	3	16322	207.79	—	—

Continúa en la siguiente página

Tabla B.4, continuación

Instancia	K	GRASP	Tiempo (s)	m-CCTP	Gap (%)
kroD100-1-49-50-2	2	20280	203.42	31665	98.85
kroD100-1-49-50-3	2	19653	199.02	—	—
kroD100-1-49-50-4	2	19922	193.06	35355	99.22
kroD100-10-40-50-1	5	35332	180.90	—	—
kroD100-10-40-50-2	4	37095	189.80	—	—
kroD100-10-40-50-3	3	40540	200.23	—	—
kroD100-10-40-50-4	3	40636	199.61	—	—
kroD100-5-20-75-1	2	18518	69.00	18675	94.96
kroD100-5-20-75-2	2	18518	63.53	18619	88.92
kroD100-5-20-75-3	2	18518	65.12	18619	89.91
kroD100-5-20-75-4	1	28445	68.83	28393	86.27

B.2 RESULTADOS OBTENIDOS EN ESTA TESIS PARA EL M-CCTP

En las Tablas B.6-B.9 se muestran los resultados computacionales obtenidos por las reformulaciones del m-CCTP propuestas en esta tesis.

En la primer columna se encuentra el nombre de la instancia, cuyas características fueron descritas en la sección anterior. La segunda y tercer columna contienen el valor de la función objetivo y el Gap (%) reportados por el optimizador CPLEX para la reformulación propuesta m-CCTP(NF). Las columnas restantes corresponden al valor de la función objetivo y el Gap (%), respectivamente, reportados por CPLEX para la reformulación propuesta fortalecida con las desigualdades válidas presentadas en la Sección 5.5 m-CCTP(NF+DV).

Cabe mencionar que, tanto en la reformulación como en la reformulación fortalecida con desigualdades, el criterio de paro utilizado por CPLEX fue 7200 s de tiempo de cómputo o 0.01 % de Gap, parando por tiempo en la mayoría de las instancias a excepción de las presentadas en la Tabla B.5 en las que el fortalecimiento con desigualdades válidas ayudó a alcanzar el óptimo.

Tabla B.5: Tiempo computacional requerido en instancias cuyo óptimo fue alcanzado por CPLEX.

Instancia	Tiempo (s)
kroA100-5-20-75-1	6292.74
kroB100-1-24-75-1	4345.72
kroB100-1-24-75-2	5866.77

Tabla B.6: Resultados computacionales obtenidos para instancias en clase A.

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroA100-1-24-75-1	12458	73.28	12240	20.04
kroA100-1-24-75-2	12240	73.05	12240	19.18
kroA100-1-24-75-3	12791	76.88	12240	16.25
kroA100-1-24-75-4	20141	84.17	20141	60.28
kroA100-1-49-50-1	19179	90.74	16947	57.92
kroA100-1-49-50-2	32354	95.84	20394	74.03
kroA100-1-49-50-3	19866	93.30	19986	69.29
kroA100-1-49-50-4	69119	98.72	40571	87.17
kroA200-1-99-100-1	—	—	26956	58.73
kroA200-1-99-100-2	—	—	29188	64.05
kroA200-1-99-100-3	—	—	34604	72.10
kroA200-1-99-100-4	39615	99.10	32843	71.03

Continúa en la siguiente página

Tabla B.6, continuación

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroA100-10-40-50-1	28803	82.71	—	—
kroA100-10-40-50-2	29424	84.55	—	—
kroA100-10-40-50-3	44563	92.15	—	—
kroA100-10-40-50-4	34298	87.82	—	—
kroA100-5-20-75-1	14899	41.29	14899	0.01
kroA100-5-20-75-2	17103	56.03	14899	9.15
kroA100-5-20-75-3	14899	53.08	14899	5.87
kroA100-5-20-75-4	27098	74.50	26571	61.65
kroA200-1-49-150-1	35063.3	95.91	21121	71.22
kroA200-1-49-150-2	26320	93.99	20220	68.95
kroA200-1-49-150-3	20357	92.50	22589	71.85
kroA200-1-49-150-4	49505	98.03	36969	84.75
kroA200-10-40-150-1	34417	85.72	27077	58.71
kroA200-10-40-150-2	33244	85.97	31719	66.06
kroA200-10-40-150-3	34968	85.69	32034	66.08
kroA200-10-40-150-4	62985	93.70	41459	76.28
kroA200-20-80-100-1	—	—	66877	82.15
kroA200-20-80-100-2	—	—	—	—
kroA200-20-80-100-3	—	—	—	—
kroA200-20-80-100-4	176558	97.97	—	—

Tabla B.7: Resultados computacionales obtenidos para instancias en clase B.

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroB100-1-24-75-1	8836	69.02	8836	0.01

Continúa en la siguiente página

Tabla B.7, continuación

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroB100-1-24-75-2	8836	68.98	8836	0.01
kroB100-1-24-75-3	12133	78.02	12133	52.32
kroB100-1-24-75-4	12133	81.76	12133	50.25
kroB100-1-49-50-1	25227	96.25	19234	71.65
kroB100-1-49-50-2	19038	93.58	18199	66.67
kroB100-1-49-50-3	28798	96.38	19682	72.89
kroB100-1-49-50-4	20223	94.07	16753	63.48
kroB200-1-99-100-1	—	—	—	—
kroB200-1-99-100-2	—	—	—	—
kroB200-1-99-100-3	—	—	—	—
kroB200-1-99-100-4	—	—	—	—
kroB100-10-40-50-1	24790	82.16	23208	57.98
kroB100-10-40-50-2	27752	85.29	26859	65.90
kroB100-10-40-50-3	27724	84.51	27204	66.19
kroB100-10-40-50-4	32081	86.84	31380	72.12
kroB100-5-20-75-1	17242	74.57	15737	44.06
kroB100-5-20-75-2	16875	71.84	15737	46.22
kroB100-5-20-75-3	16033	71.04	16238	47.57
kroB100-5-20-75-4	30728	85.22	29053	73.52
kroB200-1-49-150-1	25786.5	94.00	22661	71.11
kroB200-1-49-150-2	30328	96.17	22746	70.26
kroB200-1-49-150-3	29897	96.19	24657	74.93
kroB200-1-49-150-4	21370	94.26	23787	74.58
kroB200-10-40-150-1	—	—	36354	76.39
kroB200-10-40-150-2	—	—	39917	78.77

Continúa en la siguiente página

Tabla B.7, continuación

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroB200-10-40-150-3	50995	93.01	44713	81.45
kroB200-10-40-150-4	—	—	—	—
kroB200-20-80-100-1	—	—	—	—
kroB200-20-80-100-2	—	—	95930	85.37
kroB200-20-80-100-3	—	—	103166	87.56
kroB200-20-80-100-4	299527	98.76	169861	92.80

Tabla B.8: Resultados computacionales obtenidos para instancias en clase C.

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroC100-1-24-75-1	9529	74.28	9529	21.43
kroC100-1-24-75-2	9529	72.80	9529	37.76
kroC100-1-24-75-3	10041	73.77	9529	15.69
kroC100-1-24-75-4	9529	71.97	9529	23.42
kroC100-1-49-50-1	23941.3	92.04	15564	44.13
kroC100-1-49-50-2	20317	91.78	18263	59.30
kroC100-1-49-50-3	24881	94.43	18430	60.51
kroC100-1-49-50-4	22953	92.21	22635	65.95
kroC100-10-40-50-1	43961	88.58	32413	63.92
kroC100-10-40-50-2	33745	83.98	30047	59.42
kroC100-10-40-50-3	36254	87.08	33438	66.37
kroC100-10-40-50-4	44823	90.20	53182	81.93
kroC100-5-20-75-1	14311	48.22	14311	16.61
kroC100-5-20-75-2	14311	48.23	14311	21.32
kroC100-5-20-75-3	14311	50.37	14311	21.98

Continúa en la siguiente página

Tabla B.8, continuación

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroC100-5-20-75-4	21794	64.48	21794	55.64

Tabla B.9: Resultados computacionales obtenidos para instancias en clase D.

Instancia	m-CCTP(NF)		m-CCTP(NF+DV)	
	F. Objetivo	Gap (%)	F. Objetivo	Gap (%)
kroD100-1-24-75-1	9661	69.37	9661	25.24
kroD100-1-24-75-2	9661	64.93	10151	29.84
kroD100-1-24-75-3	16969	82.96	16641	63.29
kroD100-1-24-75-4	16969	84.51	16556	62.29
kroD100-1-49-50-1	24655	94.63	22821	71.29
kroD100-1-49-50-2	28828	96.58	30056	80.82
kroD100-1-49-50-3	26287	95.08	28525	79.73
kroD100-1-49-50-4	37556	97.20	22429	69.65
kroD100-10-40-50-1	35177	85.39	36704	72.62
kroD100-10-40-50-2	41999	90.48	37622	74.69
kroD100-10-40-50-3	49227	92.51	44976	80.56
kroD100-10-40-50-4	40620	90.25	40446	76.31
kroD100-5-20-75-1	18619	70.90	18675	57.59
kroD100-5-20-75-2	18619	73.04	18675	54.58
kroD100-5-20-75-3	18731	72.55	18619	54.22
kroD100-5-20-75-4	28393	83.93	28393	72.90

BIBLIOGRAFÍA

- [1] Renata M Aiex, Silvio Binato, and Mauricio GC Resende. Parallel grasp with path-relinking for job shop scheduling. *Parallel Computing*, 29(4):393–430, 2003.
- [2] Catherine Alexander and Chris Smaje. Surplus retail food redistribution: An analysis of a third sector model. *Resources, Conservation and Recycling*, 52(11):1290–1298., 2008.
- [3] Wilbur Olin Atwater, Charles Dayton Woods, and Arthur Peyton Bryant. *The chemical composition of American food materials*. US Government Printing Office, 1906.
- [4] Foodbank Australia. End hunger report 2013. *Foodbank Australia, Sydney*, 2012.
- [5] Christopher B. Barrett. Measuring food insecurity. *Science*, 327(5967):825–828, 2010.
- [6] Luther G. Brock and Lauren B. Davis. Estimating available supermarket commodities for food bank collection in the absence of information. *Expert Systems with Applications*, 42(7):3450 – 3461, 2015.
- [7] Carlos Carreto and Barrie Baker. A grasp interactive approach to the vehicle routing problem with backhauls. In *Essays and Surveys in Metaheuristics*, pages 185–199. Springer, 2002.
- [8] Geir Dahl and Njål Foldnes. Lp based heuristics for the multiple knapsack

- problem with assignment restrictions. *Annals of Operations Research*, 146(1): 91–104, 2006.
- [9] Lauren B. Davis, Irem Sengul, Julie S. Ivy, Luther G. Brock III, and Lastella Miles. Scheduling food bank collections and deliveries to ensure food safety and improve access. *Socio-Economic Planning Sciences*, 48(3):175 – 188, 2014.
- [10] Milind Dawande, Jayant Kalagnanam, Pinar Keskinocak, F Sibel Salman, and R Ravi. Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of combinatorial optimization*, 4(2):171–186, 2000.
- [11] FAO, WFP, and IFAD. The state of food insecurity in the world 2012: Economic growth is necessary but not sufficient to accelerate reduction of hunger and malnutrition. *Rome, FAO*, 2012.
- [12] Thomas A Feo and Mauricio GC Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67–71, 1989.
- [13] Barry E Flanagan, Edward W Gregory, Elaine J Hallisey, Janet L Heitgerd, and Brian Lewis. A social vulnerability index for disaster management. *Journal of Homeland Security and Emergency Management*, 8(1), 2011.
- [14] David A Flores-Garza, M Angélica Salazar-Aguilar, Sandra Ulrich Ngueveu, and Gilbert Laporte. The multi-vehicle cumulative covering tour problem. *Annals of Operations Research*, 258(2):761–780, 2017.
- [15] Food and Agriculture Organization of the United Nations. Human energy requirements: Report of a Joint FAO/WHO/UNU Expert Consultation, Rome: FAO, 2004.
- [16] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. WH Freeman, New York, 2002.

-
- [17] Susan Garner Garille and Saul I Gass. Stigler’s diet problem revisited. *Operations Research*, 49(1):1–13, 2001.
- [18] Pilar L González-Torre and Jorge Coque. How is a food bank managed? different profiles in Spain. *Agriculture and human values*, 33(1):89–100, 2016.
- [19] Mingyu Guo, Vincent Conitzer, and Daniel M. Reeves. *Internet and Network Economics: 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings*, chapter Competitive Repeated Allocation without Payments, pages 244–255. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [20] Minh Hoang Ha, Nathalie Bostel, André Langevin, and Louis-Martin Rousseau. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, 226(2):211–220, 2013.
- [21] Gyöngyi Kovács and Karen M Spens. Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management*, 37(2):99–114, 2007.
- [22] Lilly M. Lancaster, Joseph L. Balintfy, and Shahram Taj. Modeling the supplementing of human diets. *Socio-Economic Planning Sciences*, 39(1):1 – 10, 2005.
- [23] H Wally Lee and Marc Zbinden. Marrying logistics and technology for effective relief. *Forced Migration Review*, 18(3):34–35, 2003.
- [24] Robert W Lien, Seyed MR Iravani, and Karen R Smilowitz. Sequential resource allocation for nonprofit operations. *Operations Research*, 62(2):301–317, 2014.
- [25] Douglas Long. Logistics for disaster relief: engineering on the run. *IIE solutions*, 29(6):26–30, 1997.
- [26] Yannis Marinakis. Multiple phase neighborhood search-grasp for the capacitated vehicle routing problem. *Expert Systems with Applications*, 39(8):6807–6815, 2012.

-
- [27] Isabel Martins, Tania Guedes, Paula Rama, Joao Ramos, and Tatiana Tchémissova. Modelling the problem of food distribution by the Portuguese food banks. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2(3):313–341, 2011.
- [28] Shelley McGuire. US Department of Agriculture and US Department of Health and Human Services, Dietary Guidelines for Americans, 2010. Washington, DC: US Government Printing Office, January 2011. *Advances in Nutrition: An International Review Journal*, 2(3):293–294, 2011.
- [29] Lynn McIntyre. Food security: more than a determinant of health. *Policy Options*, 24(3):46–51, 2003.
- [30] Hitomi Okubo, Satoshi Sasaki, Kentaro Murakami, Tetsuji Yokoyama, Naoko Hirota, Akiko Notsu, Mitsuru Fukui, and Chigusa Date. Designing optimal food intake patterns to achieve nutritional goals for japanese adults through the use of linear programming optimization models. *Nutrition Journal*, 14(1):57, 2015.
- [31] Irem Sengul Orgut, Julie Ivy, Reha Uzsoy, and James R. Wilson. Modeling for the equitable and effective distribution of donated food under capacity constraints. *IIE Transactions*, 48(3):252–266, 2015.
- [32] M Rajkumar, P Asokan, N Anilkumar, and T Page. A grasp algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, 49(8):2409–2423, 2011.
- [33] Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006.
- [34] M. Angélica Salazar-Aguilar, André Langevin, and Gilbert Laporte. Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, 39(7):1432–1440, 2012.

-
- [35] Senay Solak, Christina Scherrer, and Ahmed Ghoniem. The stop-and-drop problem in nonprofit food distribution networks. *Annals of Operations Research*, 221(1):407–426, 2012.
- [36] VG Venkatesh, Rameshwar Dubey, and Sadia Samar Ali. Disaster relief operations and continuous aid program in human supply networks: are they congruent?—an analysis. In *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, pages 959–973. Springer, 2014.
- [37] B. Wilson and E. Tsoa. Hunger count 2001: food bank lines in insecure times. *Toronto: Canadian Association of Food Banks*, 2001.
- [38] World Food Summit. Rome Declaration on World Food Security and World Food Summit Plan of Action, 1996.

AUTOBIOGRAFÍA

José Álvaro Gómez Pantoja

Candidato para obtener el grado de
Doctor en Ingeniería
con Especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

MODELOS Y ALGORITMOS DE OPTIMIZACIÓN PARA LA
DISTRIBUCIÓN DE AYUDA HUMANITARIA

Nací el 27 de julio de 1991 en Dr. Arroyo, Nuevo León. Viví hasta los 15 años en la comunidad de La Ascensión, en el municipio de Aramberri, N. L., donde cursé mis estudios de primaria y secundaria.

En el año de 2006 me mudé a la ciudad de Monterrey con el fin de iniciar mis estudios de preparatoria, obteniendo una matrícula de la Universidad Autónoma de Nuevo León en la preparatoria número 9.

En 2008 comencé la Licenciatura en Matemáticas en la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León, la cual concluí en diciembre del 2012. Posteriormente, en enero de 2013, inicié la Maestría en Ciencias

con especialidad en Calidad y Productividad en el Instituto Tecnológico y de Estudios Superiores de Monterrey (TEC de Monterrey) donde participé en la Cátedra de Investigación: “Solución de Problemas de Programación Bi-nivel”, al concluir mis estudios de maestría fui premiado con la Mención Honorífica de Excelencia en la generación Ene’13-Dic’14 de dicha maestría.

En enero del año 2015 comencé mis estudios de doctorado en el Programa de Posgrado en Ingeniería de Sistemas de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León.