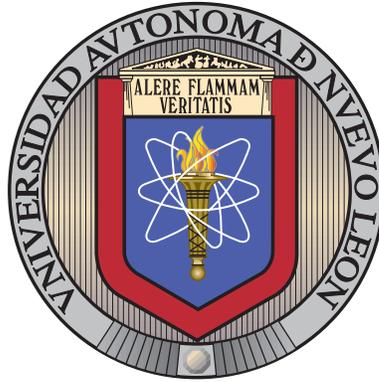


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



OPTIMIZACIÓN DEL PROBLEMA DEL p -CENTRO
CAPACITADO

POR

DAGOBERTO RAMÓN QUEVEDO OROZCO

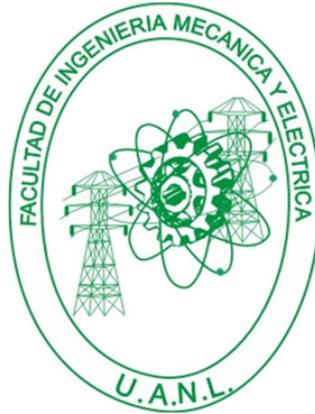
TESIS EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



OPTIMIZACIÓN DEL PROBLEMA DEL p -CENTRO
CAPACITADO

POR

DAGOBERTO RAMÓN QUEVEDO OROZCO

TESIS EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS

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 «Optimización del Problema del p -Centro Capacitado», realizada por el alumno Dagoberto Ramón Quevedo Orozco, con número de matrícula 1613610, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. Roger Z. Ríos Mercado

Director

Dr. Juan Antonio Díaz García

Revisor

Dr. Vincent Boyer

Revisor

Vo. Bo.

Dr. Simón Martínez Martínez

Subdirector

Subdirección de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, Junio 2014

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 «Optimización del Problema del p -Centro Capacitado», realizada por el alumno Dagoberto Ramón Quevedo Orozco, con número de matrícula 1613610, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



Dr. Roger Z. Ríos Mercado
Director

Dr. Juan Antonio Díaz García
Revisor



Dr. Vincent Boyer
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector
Subdirección de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, Junio 2014

OPTIMIZACIÓN DEL PROBLEMA DEL p -CENTRO
CAPACITADO

RESUMEN

OPTIMIZACIÓN DEL PROBLEMA DEL p -CENTRO CAPACITADO

Asesor: Dr. Roger Z. Ríos Mercado

Número de páginas: 112.

El problema de p -centro capacitado es un conocido problema de localización, el cual consiste en ubicar p instalaciones y realizar la asignación de usuarios a cada una de éstas, con el fin de minimizar la máxima distancia entre cualquier usuario y su instalación asignada, sujeta a la capacidad de demanda dada que cada instalación puede proporcionar. La naturaleza combinatoria del problema exige definir una metodología de solución que implique el uso de algoritmos heurísticos y técnicas avanzadas de optimización exacta que proporcionen soluciones de buena calidad en un tiempo razonable a instancias de tamaño relativamente grandes.

Este problema surge de la necesidad de reducir los efectos adversos en el peor escenario, los cuales son restringidos por una capacidad de servicio. Aplicaciones comunes pueden ser encontradas en la planificación de distritos escolares, en el diseño del sistema de cobertura en salud y más recientemente en la localización de instalaciones de emergencia donde es claro que el costo de preservar la vida humana es más importante que el costo de proveer el servicio.

El presente trabajo propone principalmente una metodología heurística para la solución de este problema, la cual integra varios componentes, tales como un método

voraz-adaptativo con una selección probabilística, búsqueda local voraz iterada y una búsqueda descendente por entornos variables. La evidencia empírica sobre un conjunto de instancias usualmente utilizadas en la literatura, basado en un diseño experimental con sustento estadístico, revela el impacto positivo de cada uno de los componentes desarrollados. Además, se encontró empíricamente que la heurística propuesta supera a la mejor heurística existente para este problema.

Se desarrolla además un método exacto de solución de ramificación, generación de columnas con planos cortantes, utilizando para su implementación computacional un marco de trabajo de reciente generación para el desarrollo de métodos de descomposición. La eficacia del método fue evaluada empíricamente.

Firma del asesor: _____

Dr. Roger Z. Ríos Mercado

ÍNDICE GENERAL

Agradecimientos	v
Resumen	vi
Índice general	viii
Índice de figuras	xii
Índice de tablas	xv
Índice de algoritmos	xvii
Simbología	xviii
1. Introducción	1
1.1. Descripción del problema	1
1.2. Antecedentes del problema	2
1.3. Motivación y justificación	4
1.4. Objetivos	5
1.5. Hipótesis	6

1.6. Estructura de la tesis	7
2. Formulación del problema	8
2.1. Definición del problema	8
2.2. Formulación de programación entera	9
2.3. Formulación de optimización combinatoria	11
2.4. Complejidad del problema	12
3. Método de solución heurístico	13
3.1. Descripción de la metaheurística propuesta	13
3.1.1. Construcción	17
3.1.2. Búsqueda local	19
3.2. Análisis de la complejidad del algoritmo	27
4. Método de solución exacto	31
4.1. Método de solución propuesto	32
4.1.1. Ramificación	33
4.1.2. Generación de cortes	34
4.1.3. Generación de columnas	35
5. Estudio computacional	40
5.1. Objetivos	40
5.2. Condiciones experimentales	41
5.2.1. Instancias de prueba	41

5.3. Evaluación del método heurístico	43
5.3.1. Calibración de parámetros	43
5.3.2. Comparación entre heurísticas	47
5.3.3. Análisis de componentes	54
5.3.4. Recursos computacionales	55
5.4. Evaluación del método exacto	57
5.4.1. Recursos computacionales	60
6. Conclusiones y contribuciones	62
6.1. Método heurístico	62
6.2. Método exacto	63
6.3. Trabajo a futuro	65
Apéndices	66
A. Marco teórico	67
A.1. Teoría de grafos	67
A.1.1. Métricas de distancia	69
A.1.2. Funciones de dispersión	70
A.2. Optimización combinatoria	71
A.2.1. Métodos de solución heurísticos	74
B. Métodos de descomposición	76
B.1. Principios básicos	77

B.2. Métodos	79
B.2.1. Dantzig-Wolfe	80
B.2.2. Lagrange	80
B.2.3. Planos cortantes	81
B.2.4. Generación de columnas con planos cortantes	82
B.3. Software	84
C. Demostración de la complejidad computacional del C_pCP	87
D. Análisis de desempeño de B&B para la solución del C_pCP	90
D.1. Condiciones y resultados	90
D.2. Conclusiones	92
E. Gráficas de análisis estadístico	94
E.1. Análisis estadístico de desviación relativa	94
E.2. Análisis del tiempo	97
Bibliografía	100
Índice alfabético	109

ÍNDICE DE FIGURAS

3.1. Fases de destrucción y reconstrucción del método de perturbación. . .	21
3.2. Ilustración de las estructuras de vecindario en el método VND.	25
3.3. Segmento de una solución que es elegible para aplicar el método de agitar.	27
5.1. Análisis estadístico de la desviación porcentual en el conjunto de ins- tancias A.	45
5.2. Análisis estadístico del tiempo en el conjunto de instancias A.	46
5.3. Comparación asintótica en función de n entre los métodos en términos de tiempo de ejecución y uso de memoria.	56
5.4. Comparación asintótica en función de n y p entre los métodos en términos de tiempo de ejecución y uso de memoria.	57
5.5. Gráfica asintótica en términos de tiempo de ejecución y uso de me- moria de BP&C.	60
A.1. Diferentes tipos de grafos.	68
A.2. Grafo planar ponderado con pesos en \mathbb{Z}_+	69
A.3. Comparación de las diferentes funciones de dispersión en un grafo planar.	71

A.4. Crecimiento de algunas funciones logarítmicas y exponenciales en función de un entero n	72
A.5. Cronograma y genealogía de las heurísticas.	74
B.1. Poliedros en un espacio \mathbb{R}^2	78
B.2. Estructura de los bloques diagonales de una matriz A'	79
B.3. Diagrama de flujo de BP&C	85
D.1. Comparación asintótica en términos de tiempo de ejecución y uso de memoria de B&B.	93
E.1. Análisis de la desviación porcentual en el conjunto de instancias A. . .	94
E.2. Análisis de la desviación porcentual en el conjunto de instancias B. . .	94
E.3. Análisis de la desviación porcentual en el conjunto de instancias C. . .	95
E.4. Análisis de la desviación porcentual en el conjunto de instancias D- α . .	95
E.5. Análisis de la desviación porcentual en el conjunto de instancias D- β . .	95
E.6. Análisis de la desviación porcentual en el conjunto de instancias D- γ . .	96
E.7. Análisis de la desviación porcentual en el conjunto de instancias D- δ . .	96
E.8. Análisis de la desviación porcentual en el conjunto de instancias E . .	96
E.9. Análisis del tiempo para el conjunto de instancias A.	97
E.10. Análisis del tiempo para el conjunto de instancias B.	97
E.11. Análisis del tiempo para el conjunto de instancias C.	97
E.12. Análisis del tiempo para el conjunto de instancias D- α	98
E.13. Análisis del tiempo para el conjunto de instancias D- β	98

E.14. Análisis del tiempo para el conjunto de instancias D- γ	98
E.15. Análisis del tiempo para el conjunto de instancias D- δ	99
E.16. Análisis del tiempo para el conjunto de instancias E.	99

ÍNDICE DE TABLAS

5.1. Resultado de la prueba de Kruskal-Wallis, respecto al nivel definido por α y la variable de respuesta DR.	45
5.2. Valores del parámetro α para cada conjunto de instancias, donde se obtienen los valores más bajos para la variable de respuesta DR. . . .	46
5.3. Comparación de los métodos en el conjunto de instancias A.	48
5.4. Comparación de los métodos en el conjunto de instancias B.	49
5.5. Comparación de los métodos en el conjunto de instancias C.	49
5.6. Comparación de los métodos en el conjunto de instancias D- α	50
5.7. Comparación de los métodos en el conjunto de instancias D- β	51
5.8. Comparación de los métodos en el conjunto de instancias D- γ	52
5.9. Comparación de los métodos en el conjunto de instancias D- δ	53
5.10. Comparación de los métodos en el conjunto de instancias E	54
5.11. Resumen de la comparación entre los métodos para todos los conjuntos de instancias.	54
5.12. Análisis de los componentes dentro de la heurística QR.	55
5.13. Resultado de BP&C en la solución del conjunto de instancias A. . . .	58
5.14. Resultado de BP&C en la solución del conjunto de instancias B. . . .	59

5.15. Resultado de BP&C en la solución del conjunto de instancias C. . . .	59
D.1. Resultado de B&B en la solución del conjunto de instancias A	91
D.2. Resultado de B&B en la solución del conjunto de instancias B	92
D.3. Resultado de B&B en la solución del conjunto de instancias C	92

ÍNDICE DE ALGORITMOS

1.	Procedimiento IGLS/VND	16
2.	Procedimiento de construcción	19
3.	Procedimiento de perturbación	20
4.	Procedimiento de búsqueda descendente por entornos variables	22
5.	Agitación de la p -partición X	26
6.	Creación del conjunto \mathcal{S}_i y vector μ_i	35
7.	Generación de columnas iniciales	39
8.	Método de generación de columnas y cortes	83

SIMBOLOGÍA

A, B, \dots	conjuntos
a, b, \dots	elementos
$\{1, \dots, n\}$	conjunto de índices de 1 a n
$ A $	cardinalidad del conjunto A
$\mathbf{c}, \mathbf{x}, \mathbf{y}$	vectores
A^\top, \mathbf{c}^\top	transpuesta
\mathbb{N}	número natural
\mathbb{Q}	número racional
\mathbb{R}	número real
\mathbb{Z}	número entero
G	grafo $G = (V, E)$
V	conjunto de nodos en un grafo
E	conjunto de aristas en un grafo
n	número de nodos
m	número de aristas
i, j, k	nodos en V
(i, j)	arista que conecta a los nodos i y j
$a \wedge b$	operador de conjunción (<i>and</i>)
$a \vee b$	operador de disjunción (<i>or</i>)
$\Omega(f(x))$	cota inferior asintótica
$\mathcal{O}(f(x))$	cota superior asintótica
$P \propto Q$	transformación polinomial de P a Q

CAPÍTULO 1

INTRODUCCIÓN

Los problemas de localización de instalaciones consisten en ubicar un conjunto de instalaciones (recursos) y asignar éstas a un conjunto de puntos de demanda o usuarios de tal manera que sea minimizado el costo de satisfacer la demanda de los usuarios con respecto a algún conjunto de restricciones [26]. La ciencia de localización de instalaciones se enfoca principalmente en las áreas de optimización discreta y continua. Los investigadores tienen la atención en algoritmos para diversos casos de estudio presentes en situaciones reales tales como: plantas industriales, bancos, comercios, servicios de salud, correos, distritos escolares, servicios postales, por nombrar algunos.

El presente trabajo aborda el estudio de un conocido problema de localización definido como el problema del p -centro capacitado ($CpCP$, por sus siglas en inglés: *capacitated vertex p -center problem*). En esta sección se abordan los aspectos básicos del problema tales como descripción general y antecedentes en la literatura así como las hipótesis, objetivos generales y específicos de este trabajo de tesis.

1.1 DESCRIPCIÓN DEL PROBLEMA

El $CpCP$ es un muy conocido problema de localización el cual consiste en localizar p instalaciones y asignar cada uno de los usuarios dados a una instalación, buscando minimizar la máxima distancia de un usuario hacia su instalación asignada. La versión básica del problema de p -centro, establece una capacidad infinita para

cada instalación, sin embargo, la versión capacitada define un valor de demanda para cada usuario y un límite en el servicio que cada instalación seleccionada puede ofrecer.

El criterio de minimax que caracteriza el objetivo del $CpCP$ es típicamente usado en aplicaciones de localización para minimizar los efectos adversos en el peor de los casos durante un servicio proporcionado a los usuarios. Aplicaciones comunes de este problema pueden ser encontradas en la planificación de distritos escolares o en el diseño del sistema de cobertura en salud. Naturalmente estos ejemplos muestran la presencia de un límite en la capacidad del servicio que cada instalación puede ofrecer. Además en algunos problemas de localización de instalaciones para emergencias, la inclusión de restricciones de capacidad puede ser usada para definir una situación de emergencia dada por proveer los servicios médicos cuidando la localización de instalaciones, ambulancias o estaciones de bomberos las cuales deberán de proveer el servicio con una pronta respuesta y rápida transportación de equipamiento. En esta situación es claro que el costo de preservar la vida humana es lo más importante [12]. Sin embargo la presencia de las restricciones de capacidad incrementa la dificultad del problema incluso sobre la versión sin capacidad del p -centro, el cual es NP-duro [48].

1.2 ANTECEDENTES DEL PROBLEMA

El problema de p -centro sin restricciones de capacidad (pCP , por sus siglas en inglés: *p-center problem*) ha sido ampliamente abordado desde un enfoque algorítmico y exacto. Los primeros estudios fueron esbozados por Hakimi [38, 39], donde se optimiza la localización de centros de conmutación en una red de comunicaciones y estaciones de policía en un sistema de carreteras y Kariv y Hakimi [48], quienes demuestran que pCP en un grafo general es NP-duro. Farahani y Hekmatfar [26] proveen un capítulo sobre el estado del arte del problema en el cual incluyen modelos, aplicaciones y métodos de solución heurísticos y exactos de los trabajos con mayor relevancia. De manera específica podemos decir que el trabajo de Elloumi et al. [24]

es uno de los trabajos más significativos para el p CP en el cual se propone una reformulación del problema basada en radios de cobertura así como una metodología de solución a través de un enfoque exacto. Desde la perspectiva heurística, Mladenović, Labbé y Hansen [63] proponen un método efectivo basado en Búsqueda Tabú (TS, por sus siglas en inglés: *tabu search*) [37] y Búsqueda por Entornos Variables (VNS, por sus siglas en inglés: *variable neighborhood search*) [42].

La investigación acerca del C_p CP ha sido más limitada. El problema fue presentado por Bar-Ilan, Kortsarz y Peleg [6] para un problema de localización de centros en redes de comunicaciones distribuidas, desarrollando un método de aproximación en tiempo polinomial cuando la demanda de cada usuario es igual a 1, es decir unitaria. Khuller y Sussmann [50] proponen otro método de aproximación, donde cada usuario tiene una demanda unitaria e introducen una variante del problema donde múltiples instalaciones pueden ser localizadas en un nodo. Denotan esta variante como «problema del múltiple p -centro capacitado». Además de estos algoritmos de aproximación, un algoritmo exacto en tiempo polinomial para árboles en redes es desarrollado por Jaeger y Goldberg [44], donde la demanda del usuario es unitaria y considerando el caso del múltiple p -centro capacitado.

Sin embargo, una situación más real se presenta cuando se tienen diferentes valores de demanda para cada usuario y diferentes valores de capacidad para un sitio potencial. Bajo este enfoque más realista, los únicos trabajos significativos desde la perspectiva de optimización exacta son proporcionados por Özsoy y Pınar [69] y Albareda-Sambola, Díaz y Fernández [4]. En el primer trabajo, los autores presentan un método exacto basado en la solución de una serie de problemas de cobertura usando un solucionador comercial de programación lineal entera-mixta (MILP, por sus siglas en inglés: *mixed integer linear programming*). En el segundo y más reciente trabajo, los autores proponen un método exacto basado en relajación Lagrangiana y una reformulación de cobertura; este trabajo es relevante por aportar cotas duales de calidad en un tiempo bastante competitivo respecto al método propuesto por Özsoy y Pınar [69]. Hasta donde se conoce en la literatura, el método heurístico más

significativo para el $CpCP$ se debe a Scaparra, Pallottino y Scutellà [85]. En este trabajo los autores desarrollan una heurística basada en una búsqueda a gran escala con un vecindario constituido de intercambios múltiples a través de una grafo de mejora, explotando los principios de la teoría de optimización de flujos en redes.

Una de las aplicaciones de los problemas de localización son los del tipo diseño territorial. Kalcsics, Nickel y Schröder [47] realizan una extensa revisión de la literatura actual de esta área. Una variante del área es la del tipo *diseño territorial* introducida por Vargas-Suárez, Ríos-Mercado y López [90]. Dentro de este contexto, entre los más recientes trabajos que presentan una función objetivo similar a la del $CpCP$ se encuentra el de Ríos-Mercado y Fernández [80], quienes proponen un método heurístico de búsqueda voraz aleatoria y adaptativa (GRASP, por sus siglas en inglés: *greedy randomized adaptive search procedure*) con mecanismos de aprendizaje reactivo para el diseño de territorios comerciales con múltiples requerimientos. Posteriormente Elizondo-Amaya, Ríos-Mercado y Díaz [23] proponen un esquema de cotas duales para minimizar la dispersión de territorios que se encuentran sujetos a varias restricciones de balance y capacidad, a través de una búsqueda binaria y relajaciones Lagrangianas.

1.3 MOTIVACIÓN Y JUSTIFICACIÓN

Como se argumentó en secciones anteriores, el $CpCP$ ha sido escasamente abordado desde la perspectiva práctica. Hasta donde se conoce en la literatura se presentan sólo tres métodos de solución, de los cuales dos son métodos exactos y el otro es una heurística. No existen instancias de prueba específicas para este problema; sin embargo, los trabajos existentes prueban la eficiencia de sus métodos utilizando tres conjuntos de instancias propuestos en diversos trabajos sobre localización, los cuales son: Biblioteca-OR [11], Galvão-ReVelle [32] y Lorena-Senne [56]. En éstos se consideran instancias con características de hasta 402 nodos y 40 instalaciones. Cabe señalar que Albareda-Sambola, Díaz y Fernández [4] introducen en su experimenta-

ción computacional al conjunto de instancias propuesto por Caselli y Righini [13], originalmente propuestas para el problema de la p -mediana capacitada ($CpMP$, por sus siglas en inglés: *capacitated p -median problem*). Alguna de estas instancias presentan la característica de que tienen un valor elevado para p respecto al número de nodos. En secciones posteriores se describirá la complejidad del problema y de qué manera el valor de p influye significativamente en el rendimiento de los métodos de solución.

Pruebas preliminares durante el planteamiento de esta tesis reflejan que los métodos conocidos no logran reportar soluciones de calidad o factibles en un tiempo razonable para diversas instancias dentro del conjunto Caselli y Righini [13]. En consecuencia el principal propósito y motivación de esta tesis es el desarrollo de un algoritmo de optimización robusto capaz de reportar soluciones factibles y de calidad en un tiempo aceptable para todos los conjuntos de instancias conocidos, enfocado principalmente hacia la perspectiva de métodos heurísticos. Como justificación principal y como fue mencionado con anterioridad, el único método heurístico conocido en la literatura se debe a Scaparra, Pallottino y Scutellà [85], el cual no logra reportar soluciones factibles para un gran número de instancias. Otra motivación importante se encuentra en el hecho de que en la práctica es posible formular instancias con las características antes mencionadas, por lo que es necesario enriquecer la literatura existente con métodos de solución alternativos que logren aportar una solución factible y de utilidad en la práctica.

1.4 OBJETIVOS

Este trabajo propone derivar una metodología de solución aproximada para el $CpCP$. La naturaleza combinatoria del problema exige definir estrategias que impliquen el uso de algoritmos heurísticos robustos que proporcionen soluciones de buena calidad en un tiempo razonable. De manera exploratoria se planea abordar la solución por métodos exactos que en la actualidad son estado del arte para problemas de

programación entera como el método de ramificación y generación de columnas con planos cortantes (BP&C, por sus siglas en inglés: *branch-price & cut*). A continuación se definen los objetivos específicos:

- Desarrollar un método de aproximación robusto basado en metaheurísticas, de tal manera que se logren soluciones factibles y de calidad en un tiempo aceptable para todos los conjuntos de instancias de prueba conocidos.
- Realizar un análisis algorítmico con la finalidad de establecer el orden de operaciones requerido por el método desarrollado.
- Evaluar el desempeño del método desarrollado y cada uno de sus componentes con base en un diseño experimental adecuado así como captar los parámetros que nos ayuden a obtener soluciones de mejor calidad. Esta evaluación empírica se respalda con su correspondiente análisis estadístico.
- Realizar una comparativa entre todos los métodos encontrados en la literatura para este problema, realizando un análisis asintótico basado en calidad de la solución y recursos computacionales.
- Realizar investigación preliminar sobre el método de solución exacta BP&C que incluya la derivación de un método de solución, su implementación computacional basado en el sistema de librerías COIN-OR y una evaluación empírica preliminar.

1.5 HIPÓTESIS

Hipótesis 1 *El desarrollo de un método de aproximación robusto basado en metaheurísticas puede proporcionar soluciones factibles y de calidad en un tiempo aceptable para los diversos conjuntos de instancias de prueba disponibles para CpCP respecto al único método de solución heurístico propuesto.*

Hipótesis 2 *El desarrollo de un método exacto basado en BP&C puede proporcionar soluciones factibles y en menor tiempo para todos los diversos conjuntos de instancias de prueba disponibles para el CpCP respecto a los métodos de solución exactos propuestos.*

1.6 ESTRUCTURA DE LA TESIS

La estructura del presente trabajo está dada de la manera siguiente. En el Capítulo 2 se discuten los modelos de optimización del problema, abordando la versión como un programa entero mixto lineal y su correspondiente modelo de optimización combinatoria, además de definir supuestos, notaciones y la complejidad computacional. El Capítulo 3 describe a detalle el método de solución heurístico propuesto, así como un análisis algorítmico de cada uno sus componentes. En el Capítulo 4 se definen las bases y detalles de implementación de un método exacto basado en BP&C. En el Capítulo 5 se describen las instancias de prueba y se muestra los resultados empíricos. En éstos se presentan el ajuste de parámetros, una completa comparación entre los métodos existentes, un análisis de los componentes que conforman la heurística y la experimentación del método exacto propuesto. En el Capítulo 6 se presentan las conclusiones y aportaciones. Finalmente en los Apéndices A y B se describen las bases teóricas y notaciones relacionadas con esta tesis.

CAPÍTULO 2

FORMULACIÓN DEL PROBLEMA

En este capítulo se describe a mayor detalle la definición del problema y las formulaciones existentes para el CpCP, además se definen los supuestos generales así como la nomenclatura que se utilizará en secciones subsecuentes. Se presentan las formulaciones MILP y el modelo de optimización combinatoria. La razón por la cual se muestran ambos tipos de modelos radica en que la formulación y nomenclatura combinatoria proporciona una mayor claridad cuando se trabaja con métodos de solución heurísticos, en contraste con la formulación MILP que es de mayor utilidad cuando se desea definir un método de solución exacto.

2.1 DEFINICIÓN DEL PROBLEMA

El problema CpCP es un muy conocido problema dentro de la teoría de localización. El problema se define de la manera siguiente. Sea $U = \{u_1, \dots, u_m\}$ los puntos potenciales para localización de las instalaciones y $V = \{v_1, \dots, v_n\}$ el conjunto de usuarios. La presente tesis asume el caso especial cuando el conjunto de instalaciones potenciales coincide con el conjunto de demanda, es decir $U = V$. Para cada nodo $i, j \in V$ se conocen la demanda w_j y la capacidad de la ubicación potencial s_i ; la distancia d_{ij} entre cada par de nodos $i, j \in V$; y el número de instalaciones p . Se debe decidir dónde ubicar las p instalaciones y cómo asignar cada usuario a cada una de ellas, buscando minimizar la máxima distancia de un usuario hacia su instalación asignada.

Los modelos existentes para el C_p CP toman en cuenta los siguientes requerimientos y suposiciones:

- El conjunto de nodos V representa a su vez los usuarios y los puntos potenciales para la localización de instalaciones, por lo que se asume el caso especial cuando $U = V$.
- Se asume que el grafo es completo, es decir que existe una conexión entre cualquier par de nodos $i, j \in V$.
- Todos los usuarios deberán ser asignados estrictamente a sólo una instalación.
- El total de la demanda de los usuarios asignados a un nodo $i \in V$ designado como instalación, no deberá exceder la capacidad del nodo delimitada por s_i .
- El número de instalaciones a ser aperturadas es un valor fijo, el cual es estrictamente igual a p .

2.2 FORMULACIÓN DE PROGRAMACIÓN ENTERA

A continuación se presenta el modelo MILP abordado en la literatura acerca del C_p CP [85, 69, 4], el cual se deriva del clásico modelo del p CP, incluyendo una ampliación de restricciones de capacidad.

Índices y conjuntos

V = Conjunto de nodos, que representa instalaciones potenciales y usuarios,

n = Número de nodos,

i = Nodo de las instalaciones potenciales; $i \in V$,

j = Nodo de los usuarios; $j \in V$.

Parámetros

p = Número de instalaciones,

d_{ij} = Distancia entera entre i y j ; $i, j \in V$,

s_i = Capacidad de la instalación potencial $i \in V$,

w_j = Demanda del usuario $j \in V$.

Variables de decisión

$$x_{ij} = \begin{cases} 1, & \text{Si el usuario } j \text{ es asignado a la instalación localizada en el nodo } i; \\ 0, & \text{en otro caso.} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{Si la instalación es localizada en el nodo } i; \\ 0, & \text{en otro caso.} \end{cases}$$

Modelo matemático

$$(CpCP) \ v = \text{Minimizar } z \tag{2.1}$$

$$\text{s. a. } \sum_{i \in V} d_{ij} x_{ij} \leq z \quad j \in V, \tag{2.2}$$

$$\sum_{i \in V} x_{ij} = 1 \quad j \in V, \tag{2.3}$$

$$\sum_{j \in V} w_j x_{ij} \leq s_i y_i \quad i \in V, \tag{2.4}$$

$$\sum_{i \in V} y_i = p, \tag{2.5}$$

$$x_{ij}, y_i \in \{0, 1\} \quad i \in V, j \in V. \tag{2.6}$$

Las expresiones (2.1) y (2.2) representan la linealización de la función objetivo original $v = \min_{i \in V} \max_{j \in V} \{d_{ij} x_{ij}\}$. Las restricciones (2.2) aseguran que cada usuario es asignado exactamente a una única instalación. Las restricciones (2.3) establecen para cada instalación, que la demanda total de los usuarios asignados a ella no excede

su capacidad. La restricción (2.5) establece que el número de instalaciones que se habilitarán es igual a p . Finalmente, las restricciones (2.6) indican la naturaleza de las variables x_{ij} y y_i . La función objetivo de minimax que caracteriza al CpCP, es frecuentemente usada en modelos de localización donde es requerido minimizar los efectos adversos en el peor de los casos para satisfacer la demanda de los usuarios.

2.3 FORMULACIÓN DE OPTIMIZACIÓN COMBINATORIA

Se propone el siguiente modelo de optimización combinatoria, cuya nomenclatura será de gran utilidad durante la definición de los componentes del método de solución heurístico. Una p -partición de V es denotada por $X = \{X_1, \dots, X_p\}$ con el conjunto de índices de las particiones dado por $K = \{1, \dots, p\}$, donde $X_k \subset V$ es llamado un subconjunto de V . Cada X_k es formado por un subconjunto de nodos tal que $\bigcup_{k \in K} X_k = V$ y $X_k \cap X_l = \emptyset$ para todo $k, l \in K, k \neq l$. El conjunto de centros es denotado por $P \subset V$ tal que $P = \{c(1), \dots, c(k)\}$ donde $c(k)$ es el nodo activo como instalación para X_k , el cual es calculado, tomando en cuenta la capacidad, de la manera siguiente

$$c(k) = \arg \min_{i \in X_k} \left\{ \max_{j \in X_k} \left\{ d_{ij} : \sum_{j' \in X_k} w_{j'} \leq s_i \right\} \right\}. \quad (2.7)$$

El problema entonces, puede ser representado por el siguiente modelo combinatorio

$$\min_{X \in \Pi} \max_{k \in K} f(X_k), \quad (2.8)$$

donde Π es la colección de todas las p -particiones posibles de V . Para un subconjunto X_k , su costo asociado, también llamado costo de cuello de botella está dado por la función $f(X_k) = \max_{j \in X_k} \{d_{jc(k)}\}$. Por convención definimos que para un X_k dado, si no hay un $i \in X_k$ tal que $\sum_{j \in X_k} w_j \leq s_i$, entonces $f(X_k) = \infty$, y por lo tanto la configuración o solución definida en la p -partición es infactible. Finalmente $\ell(j) = k \in K : j \in X_k$, denota el índice de la partición $k \in K$ que suministra al nodo j .

2.4 COMPLEJIDAD DEL PROBLEMA

Como se abordó en secciones anteriores, el $CpCP$ resulta ser una variante directa del pCP . Desafortunadamente, la presencia de las restricciones de capacidad aumenta considerablemente la dificultad del problema, incluso aun más que la versión sin capacidad, la cual se sabe es NP-duro [48]. En el Apéndice C se realiza la demostración de complejidad computacional del $CpCP$.

Una manera práctica de demostrar la dificultad en la solución de un problema combinatorio es resolverlo a través de un método exacto clásico y de propósito general tal como lo es el método de ramificación y acotamiento (B&B, por sus siglas en inglés: *branch and bound*) realizando la implementación computacional a través de ILOG CPLEX.

En el Apéndice D se realiza un análisis de desempeño de B&B en su aplicación al $CpCP$. El objetivo principal de este análisis fue conocer el alcance de B&B en función del tamaño de la instancia, evaluando además la calidad de la solución y el tiempo y memoria empleado por el método. Como resultado de la evaluación empírica, se concluye que para instancias de hasta 100 nodos y 5 a 10 instalaciones a ubicar es posible encontrar soluciones óptimas para el $CpCP$ de una manera eficiente. Sin embargo para instancias de mayor tamaño no fue posible encontrar soluciones de calidad (desviación entre sus cotas) aceptables después de satisfacer los criterios de paro (3,600 segundos y 4 GB de memoria). Debido a que el tamaño de las instancias utilizadas en el presente trabajo tienen un tamaño de hasta 3080 nodos y un número de hasta 1000 instalaciones a ser ubicadas, B&B no representa una herramienta de solución viable para el problema.

CAPÍTULO 3

MÉTODO DE SOLUCIÓN HEURÍSTICO

En este capítulo se define a detalle la estrategia de solución propuesta para el CpCP desde un enfoque heurístico. Cada componente de la heurística es descrito, justificado e ilustrado algorítmicamente. Se definen los criterios de evaluación en la calidad de la solución y finalmente se realiza un análisis de la complejidad computacional del método propuesto.

3.1 DESCRIPCIÓN DE LA METAHEURÍSTICA PROPUESTA

En años recientes, una importante tendencia en el campo de las metaheurísticas ha sido la integración de diferentes componentes que ha propiciado el desarrollo de exitosos métodos híbridos que permiten explotar la estructura específica del problema [35]. Una consecuencia de esto es que, en ocasiones, no resulta del todo claro el nombre de la heurística, ya que está integrada por componentes que provienen de diferentes marcos. Con respecto al método heurístico, para el problema en estudio se ha propuesto un método que usa una búsqueda local voraz iterada (IGLS, por sus siglas en inglés: *iterated greedy local search*) como marco principal. En la fase de construcción se implementa una estrategia basada en un método voraz-adaptativo con una selección probabilística, seguido de una fase de mejora compuesta por una hibridación de IGLS y un búsqueda descendente por entornos variables (VND, por sus siglas en inglés: *variable neighborhood descent*) con estructuras de vecindario de intercambio e inserción.

Los métodos voraces-adaptativos con selección probabilística, son una variante de los métodos voraces convencionales. La idea es construir de forma iterativa una solución, pero usando una probabilidad de selección de los elementos que varía conforme la búsqueda avanza, es decir, es adaptativa al entorno. Entre los trabajos más significativos con este paradigma se encuentra Haouari y Chaouachi [43], quienes proponen un método para la solución del problema de cobertura de conjuntos (SCP, por sus siglas en inglés: *the set covering problem*). Recientemente, Amiri [5] propone un método de solución para el problema en la asignación de canales de radio en las redes celulares.

IGLS es una técnica originalmente propuesta por Ruiz y Stützle [82, 83] que resulta de una extensión de una búsqueda local iterada (ILS por sus siglas en inglés: *iterated local search*) [57]. IGLS aplica iteradamente una destrucción, reconstrucción y búsqueda local de una solución dada, enfocándose en el espacio de soluciones localmente óptimas. Con el fin de escapar de un óptimo local y explorar otras regiones en el espacio de soluciones, IGLS aplica un proceso de perturbación para generar nuevos puntos de partida para la búsqueda local, modificando la solución actual. La destrucción/construcción o perturbación, consiste en remover algunos elementos de la solución actual bajo un criterio específico, seguido por una reasignación de estos elementos a la solución a través de un método voraz. Se pueden obtener mejores resultados si durante la perturbación se tienen en cuenta las propiedades del problema. El último paso consiste en decidir, basados en un criterio de mejora, si la solución obtenida después de la búsqueda local deberá reemplazar la solución incumbente. IGLS itera hasta que el criterio de paro es satisfecho.

Una ventaja de IGLS es que permite diversificar y mejorar a lo largo de la búsqueda sin necesidad de emplear estructuras de memoria compleja, su simplicidad permite que sea aplicable a varios problemas de optimización combinatoria. Por nombrar algunos, Ruiz y Stützle [82, 83], Fanjul-Peyro y Ruiz [25], Urlings, Ruiz y Stützle [88] proporcionan resultados que son estado del arte para diferentes problemas de secuenciación.

VND [41, 35] es un algoritmo iterativo de mejora, cuya idea general es similar a una búsqueda en entornos variables (VNS, por sus siglas en inglés: *variable neighborhood search*) propuesta por Hansen y Mladenović [40]. En VND, t vecindarios son utilizados, los cuales son típicamente ordenados según su tamaño. El algoritmo inicia con el primer vecindario y realiza iteradamente la etapa de mejora hasta que el óptimo local es encontrado. Cuando no es encontrada una mejora para el h -ésimo vecindario y $h + 1 \leq t$, entonces VND continúa la búsqueda en el $(h + 1)$ -ésimo vecindario. Si una mejora en el objetivo es encontrada, la búsqueda es reiniciada en el primer vecindario. Se ha demostrado en la literatura que VND puede mejorar considerablemente el rendimiento de los algoritmos iterativos de mejora, respecto a la calidad de la solución final y el tiempo requerido para encontrar soluciones de buena calidad comparado con otros métodos iterativos utilizados en vecindarios a gran escala [40].

La idea de hibridación de ILS con VND/VNS ha sido exitosamente usada anteriormente para otros problemas de optimización combinatoria. Ribeiro et al. [79] implementan una variante para resolver un caso real de secuenciación de automóviles. Subramanian et al. [87] usan una versión paralela para el problema de rutas de vehículos (VRP, por sus siglas en inglés: *vehicle routing problem*) con recolección y entrega simultánea; Martins et al. [62] proponen un método para resolver un problema de enrutamiento y asignación de longitud de onda para redes ópticas. Otra ventaja interesante respecto a IGLS es que permite diversificar y mejorar la solución durante el trayecto de la búsqueda sin necesidad de implementar complejas estructuras de memoria.

El método propuesto se muestra en el Algoritmo 1. Con los pasos 1–2 se obtiene una solución inicial. Dentro del ciclo principal (pasos 4–13), la búsqueda local (pasos 6–7) se realiza mientras el criterio de paro no es satisfecho. Para discernir si una solución es mejor que otra, se utiliza un criterio efectivo de mejora propuesto por

Scaparra, Pallottino y Scutellà [85] definido de la siguiente forma:

$$X' \text{ es mejor que } X \Leftrightarrow \begin{cases} f(X') < f(X), \text{ o} \\ f(X') = f(X), \mathcal{B}(X') \subseteq \mathcal{B}(X), \mathcal{J}(X') \subset \mathcal{J}(X), \end{cases} \quad (3.1)$$

donde $\mathcal{B}(X)$ denota el conjunto de particiones «cuello de botella» en X definido como $\mathcal{B}(X) = \{k \in K : f(X_k) = f(X)\}$ y $\mathcal{J}(X)$ contiene a los nodos de demanda con una máxima distancia desde una instalación activa en cada partición X_k , es decir $\mathcal{J}(X) = \{j \in X_k : d_{j_c(k)} = f(X), k \in \mathcal{B}(X)\}$. El criterio (3.1) es satisfecho si decrece el valor de la función objetivo o reduce el número de particiones «cuello de botella» mientras el valor de la función objetivo no sea empeorado y sin crear nuevas particiones y nodos de demanda «cuello de botella». La solución incumbente X se actualiza si se encuentra una mejor solución factible X' de acuerdo con el criterio (3.1), en otro caso se realiza una agitación de la solución X' (paso 10). El método se detiene cuando se alcanza un número máximo de iteraciones (denotado por $r_{\text{máx}}$). Cada uno de los componentes se discute e ilustra algorítmicamente en los apartados siguientes.

Algoritmo 1 Procedimiento IGLS/VND

Require: $V = \{1, \dots, n\}$, $p \in \mathbb{N}$, $\alpha \in [0.0, 1.0]$ y $r_{\text{máx}} \in \mathbb{N}$

- 1: $X' \leftarrow \text{Construccion}(V, p)$
 - 2: $X' \leftarrow \text{VND}(X')$
 - 3: $X \leftarrow X'$
 - 4: **while** $r_{\text{máx}} > 0$ **do**
 - 5: $X' \leftarrow \text{Perturbacion}(X', \alpha)$
 - 6: $X' \leftarrow \text{VND}(X')$
 - 7: **if** X' es mejor que X **then**
 - 8: $X \leftarrow X'$
 - 9: **else**
 - 10: $X' \leftarrow \text{Agitar}(X')$
 - 11: **end if**
 - 12: $r_{\text{máx}} \leftarrow r_{\text{máx}} - 1$
 - 13: **end while**
 - 14: **return** X
-

3.1.1 CONSTRUCCIÓN

La fase de construcción consiste de dos etapas: (a) localización de instalaciones o centros y (b) asignación de usuarios. La primera etapa usa una estrategia basada en método voraz aleatorizado y adaptativo con una selección probabilística la cual diversifica el conjunto de centros potenciales. En la segunda etapa, se utiliza un método voraz determinista con un criterio de distancia y capacidad para realizar la asignación de los usuarios.

ETAPA A: LOCALIZACIÓN DE CENTROS

El objetivo de esta fase es seleccionar un conjunto adecuado de p centros o nodos iniciales. La selección se realiza a través de un proceso constructivo voraz aleatorizado y adaptativo, tomando en cuenta el factor de distancia y la capacidad de cada nodo $j \in V$. Como se sabe, en un método voraz es importante definir una función de mérito apropiada. En el caso de nuestro problema en particular, se incorpora una función voraz basada en la función propuesta por Dyer y Frizie [21] para el p CP. La fase de localización inicia seleccionando el primer centro i^* aleatoriamente. Entonces, iteradamente se selecciona el siguiente centro buscando un nodo cuya distancia respecto a los centros previamente seleccionados sea la mayor. La motivación de esto es tratar de obtener centros que estén lo más dispersos posibles, pero también favoreciendo una selección de centros con una mayor capacidad, de tal manera que durante la segunda fase sean asignados un mayor número de usuarios, reduciendo la incidencia de una solución inicial infactible. Esto es definido formalmente de la manera siguiente. Sea $P \subset V$ un conjunto parcial de centros seleccionados. Entonces, para cada $j \in V \setminus P$ se calcula la siguiente función voraz

$$\gamma(j) = s_j d_{i^*j}. \quad (3.2)$$

El siguiente nodo i^* es seleccionado usando un valor de probabilidad determinado por la función (3.2). La probabilidad normal, definida por $\pi(j)$, de seleccionar el

nodo $j \in V$ puede ser computada como

$$\pi(j) = \frac{\gamma(j)}{\sum_{j' \in V \setminus P} \gamma(j')}. \quad (3.3)$$

El uso de esta función para asignar una probabilidad de elegir el nodo $j \in V \setminus P$ favorece a los candidatos con mejor evaluación de la función (3.2). Además, una selección aleatoria con probabilidad $\pi(j)$ permite diversificar significativamente la selección inicial del conjunto P . Una vez que el nodo seleccionado es añadido al conjunto P , se calcula nuevamente la probabilidad de selección de los nodos $j \in V \setminus P$. El proceso se repite iteradamente hasta que el conjunto P tiene p nodos, es decir $|P| = p$.

ETAPA B: ASIGNACIÓN DE USUARIOS

Una vez que los centros son establecidos en el conjunto P , la segunda etapa consiste en asignar los usuarios a cada uno de los centros. Esta etapa es realizada de manera voraz determinista. Pruebas preliminares mostraron que usando una estrategia voraz aleatorizada no se aporta una mejora sustancial en la calidad de la solución, por lo que una estrategia puramente voraz y determinista resulto ser más eficiente. Es claro que el conjunto de usuarios restantes por asignar está dado por $V \setminus P$. Para este fin, se define una función voraz que mide el costo de asignar un usuario j al centro k localizado en $c(k)$ de la manera siguiente

$$\phi(j, k) = \max \left\{ \frac{d_{jc(k)}}{\bar{d}}, -r(k) + w_j \right\}, \quad (3.4)$$

donde $\bar{d} = \max_{i,j \in V} \{d_{ij}\} + 1$ es un factor de normalización y

$$r(k) = s_{c(k)} - \sum_{j' \in X_k} w_{j'}, \quad (3.5)$$

es la capacidad residual para el centro k localizado en $c(k)$. La motivación de esta función es la siguiente. Si la restricción de capacidad es satisfecha, la función (3.4) sólo toma en cuenta el factor de la distancia, en otro caso la función retorna un valor entero que penaliza la asignación. Entonces cada nodo j es asignado al centro que minimice el valor de la función (3.4), es decir $X_{k^*} \leftarrow X_{k^*} \cup \{j\}$ donde $k^* = \arg \min_{k \in K} \phi(j, k)$. Finalmente, una vez que la asignación ha sido realizada, los centros de la partición son actualizados usando la función (2.7). El Algoritmo 2 representa el método de construcción.

Algoritmo 2 Procedimiento de construcción

Require: $V = \{1, \dots, n\}$ y $p \in \mathbb{N}$

```

1:  $P \leftarrow \emptyset$ 
2: Elegir  $i^* \in V$  aleatoriamente
3:  $P \leftarrow P \cup \{i^*\}$ 
4: while  $|P| < p$  do
5:   Actualizar  $\gamma(j)$  y  $\pi(j)$ ,  $j \in V \setminus P$ , de acuerdo a (3.2) y (3.3)
6:   Elegir  $i^* \in V \setminus P$  aleatoriamente usando una probabilidad  $\pi(j)$ 
7:    $P \leftarrow P \cup \{i^*\}$ 
8: end while
9:  $X_k \leftarrow \{P_k\}$ ,  $k \in K$ 
10: for all  $j \in V \setminus P$  do
11:    $k^* \leftarrow \arg \min_{k \in K} \phi(j, k)$ 
12:    $X_{k^*} \leftarrow X_{k^*} \cup \{j\}$ 
13:   Actualizar  $c(k^*)$ 
14: end for
15: return  $X$ 

```

3.1.2 BÚSQUEDA LOCAL

Para la mejora de la solución, la fase de búsqueda local utiliza una VND dentro de un marco de IGLS, la cual es realizada en los pasos 4–13 del Algoritmo 1. En esta fase, la perturbación consiste en desconectar y reasignar usuarios de su centro, de acuerdo con la idea del IGLS, el cual se efectúa en el paso 5. Después se realiza el método de VND (paso 6), donde se usan dos diferentes estructuras de vecindarios. Finalmente si la calidad de la solución no es mejorada (paso 7) se usa un proceso agresivo de destrucción/reconstrucción parcial de la solución (paso 10). Cada uno de los componentes de esta etapa se definen a continuación.

PROCEDIMIENTO DE PERTURBACIÓN

Este método toma una solución como entrada y aplica los pasos de destrucción y reconstrucción. El objetivo específico de este procedimiento es reducir los elementos «cuello de botella» utilizando un criterio de desconexión de nodos para cada partición. El método es presentado en el Algoritmo 3.

Algoritmo 3 Procedimiento de perturbación

Require: p -partición X y $\alpha \in [0.0, 1.0]$

```

1:  $W \leftarrow \emptyset$ 
2: for all  $k \in K$  do
3:   Actualizar  $\rho(j), j \in X_k \setminus \{c(k)\}$ 
4:   while  $|W|/|X_k| - 1 < \alpha$  do
5:     Elegir  $j \in X_k \setminus \{c(k)\}$  aleatoriamente usando la probabilidad  $\rho(j)$ 
6:      $W \leftarrow W \cup \{j\}$ 
7:   end while
8: end for
9: Ordenar  $W$  de mayor a menor  $d_{j\ell(j)}, j \in W$ 
10:  $X \leftarrow X \setminus W$ 
11: for all  $j' \in W$  do
12:    $k^* \leftarrow \arg \min_{k \in K} \phi(j', k)$ 
13:    $X_{k^*} \leftarrow X_{k^*} \cup \{j'\}$ 
14: end for
15: Actualizar  $c(k), k \in K$ 
16: return  $X$ 

```

En este caso, para una partición X_k , la fase de destrucción (paso 1–10) es realizada desconectando el α % de nodos localizados en X_k (pero excluyendo al centro localizado en $c(k)$), con altos valores de la función de probabilidad definida por

$$\rho(j) = \frac{d_{jc(k)}}{\sum_{j' \in X_k \setminus \{c(k)\}} d_{j'c(k)}}. \quad (3.6)$$

La elección de este criterio de selección es motivada por el hecho de que los nodos lejanos de su centro son los que afectan en mayor grado a la función de dispersión, es decir, alguno de los nodos desconectados pertenece al conjunto de nodos «cuello de botella» $\mathcal{J}(X)$, expresado de otra forma $\mathcal{J}(X) \subset W$. La fase de reconstrucción reasigna cada nodo desconectado al centro que minimice la función (3.4), esto es

$X_{k^*} \leftarrow X_{k^*} \cup \{j\}$ donde $k^* = \arg \min_{k \in K} \phi(j, k)$. Una prioridad de asignación es dada a los nodos «cuello de botella», es decir, nodos cuya asignación previa coincidió o se aproximó al valor de la función objetivo, lo cual se hace ordenando el conjunto de nodos desconectados W (paso 9) de acuerdo a la distancia con el centro previamente asignado. Finalmente, cuando la asignación es realizada, los centros de la p -partición son actualizados utilizando la función (2.7). Las fases de destrucción y reconstrucción del método de perturbación son ilustradas en la Figura 3.1.

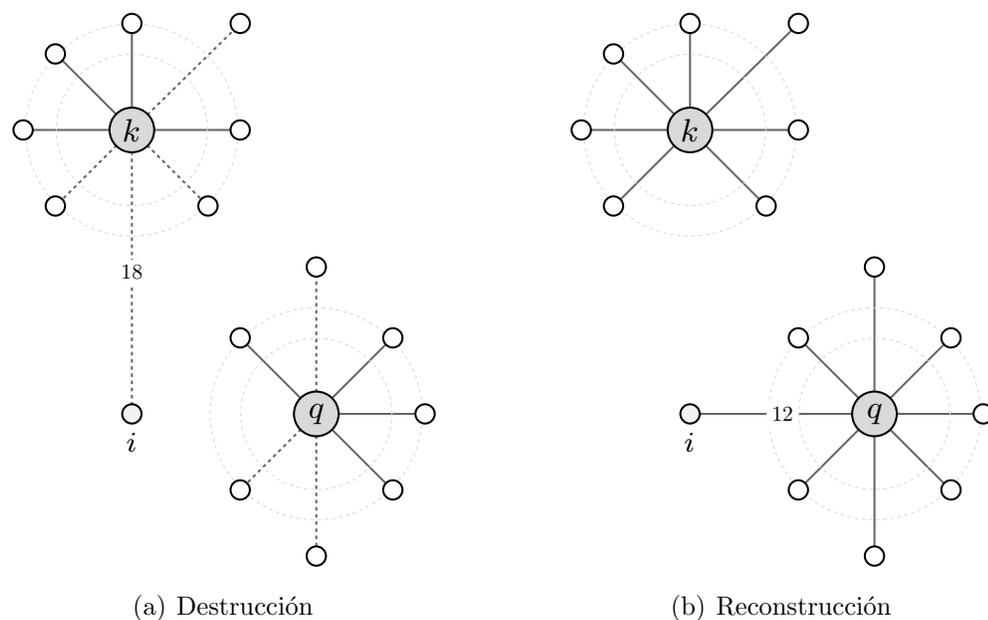


Figura 3.1: Fases de destrucción y reconstrucción del método de perturbación.

En este ejemplo, los nodos k y q representan los centros de las dos particiones. El nodo i es un elemento «cuello de botella» o el nodo que genera el peor costo de la solución actual. En este caso, antes de la destrucción la función objetivo tiene un costo $f(X) = 18$. En la fase de destrucción los nodos con línea punteada representan el conjunto de nodos a ser desconectado de su centro basados en una probabilidad $\rho(j)$. En la fase de reconstrucción, el nodo i tiene una mayor prioridad para la reasignación y cambia su localización de k a q , reduciendo el costo de la solución actual a doce.

Es importante puntualizar que una mejora en el valor de la función objetivo no necesariamente sucede durante la perturbación. Esto es debido al hecho de que algunos nodos podrían empeorar el valor de objetivo de la solución durante la fase de reconstrucción, consecuencia de las restricciones de capacidad. Sin embargo esto no es precisamente un mal efecto, por que esto permite diversificar la solución actual que posteriormente puede ser mejorada en la etapa del VND.

Como es usual en IGLS, los parámetros que intervienen en la fase de destrucción, en este caso α , juegan un importante papel en la calidad de la solución encontrada por el método. Los valores apropiados del parámetro α son ajustados con un procedimiento empírico y justificado de forma estadística para un número considerable instancias en el Capítulo 5.

BÚSQUEDA DESCENDENTE POR ENTORNOS VARIABLES

La implementación de VND propuesta se muestra en el Algoritmo 4, la cual emplea dos diferentes estructuras de vecindario denotadas por \mathcal{N}_1 y \mathcal{N}_2 . Para cada nodo $i \in \mathcal{J}(X)$ se aplica el mejor movimiento del h -ésimo vecindario (paso 2). Para cada uno de los vecindarios, el movimiento potencial toma en cuenta los factores de distancia y capacidad.

Algoritmo 4 Procedimiento de búsqueda descendente por entornos variables

Require: p -partición X

```

1: while  $h \leq 2$  do
2:    $X' \leftarrow X$ 
3:    $X' \leftarrow \mathcal{N}_h(i, X'), i \in \mathcal{J}(X)$ 
4:   Actualizar en  $X'$   $c(k), k \in K$ 
5:   if  $X'$  es mejor que  $X$  then
6:      $X \leftarrow X'$ 
7:      $h \leftarrow 1$ 
8:   else
9:      $h \leftarrow h + 1$ 
10:  end if
11: end while
12: return  $X$ 

```

Por la naturaleza descendente de la función objetivo, tiene perfecto sentido sólo considerar un subconjunto de movimientos prometedores, no todo el vecindario, es decir: sólo considerar movimientos donde la distancia entre i y una potencial partición destino en $c(k)$ es estrictamente menor que $f(X)$ y que la capacidad no sea excedida en ningún centro involucrado en el movimiento. Estas restricciones descartan movimientos no factibles, reduciendo el tamaño del vecindario a ser explorado, y por lo tanto incrementando el rendimiento del proceso. Una estrategia de mejor movimiento encontrado (*best improving*) es adoptada a través de la búsqueda. Una vez que ha sido aplicado el movimiento del h -ésimo vecindario, se actualizan los centros de la p -partición X' (paso 4). Finalmente, se actualiza la solución incumbente si se satisface el criterio de mejora definido en (3.1). A continuación se describe cada uno de los vecindarios y los criterios de evaluación.

\mathcal{N}_1) *Reinserción*: Se consideran movimientos donde un nodo $j \in \mathcal{J}(X)$ (actualmente asignado al centro de la partición X_k) es asignado a la partición X_q , tal que $k \neq q$, es decir dada una p -partición X el movimiento es definido por

$$m_1(i, q) = \{X_1, \dots, X_k \setminus \{i\}, \dots, X_q \cup \{i\}, \dots, X_p\}, \quad (3.7)$$

el movimiento es válido si y sólo si éste no excede la capacidad de la partición destino q y la distancia entre i y $c(q)$ es estrictamente menor a $f(X)$, es decir

$$r(q) \geq w_i, d_{ic(q)} < f(X). \quad (3.8)$$

La evaluación de cada movimiento se realiza usando una función $\psi_1(i, q)$ que toma en cuenta un criterio de distancia, de la manera siguiente

$$\psi_1(i, q) = d_{ic(k)} - d_{ic(q)}, \quad (3.9)$$

esta función obtiene altos valores cuando la distancia entre i y $c(q)$ es menor que la asignación actual i a $c(k)$. Entonces, se aplica la configuración de movimiento con la mejor evaluación de esta función.

\mathcal{N}_2) *Intercambio*: Se consideran movimientos donde un nodo $i \in \mathcal{J}(X)$ (actualmente asignado al centro de la partición X_k) y $j \in X_q \setminus \{c(q)\}$, con $k \neq q$ son intercambiados, es decir dada una p -partición X el movimiento es definido por

$$m_2(i, j) = \{X_1, \dots, X_q \cup \{j\} \setminus \{i\}, \dots, X_k \cup \{i\} \setminus \{j\}, \dots, X_p\}, \quad (3.10)$$

el movimiento es válido si y sólo si no se excede la capacidad de las respectivas particiones destino y la distancia entre i y $c(q)$, j y $c(k)$ es estrictamente menor que $f(X)$, es decir

$$r(k) - w_i \geq w_j, r(q) - w_j \geq w_i, d_{jc(k)} < f(X), d_{ic(q)} < f(X). \quad (3.11)$$

La evaluación de cada movimiento factible se realiza usando una función $\psi_2(i, q)$ que toma en cuenta un criterio de distancia, de la manera siguiente

$$\psi_2(i, j) = (d_{ic(k)} - d_{ic(q)}) + (d_{jc(q)} - d_{jc(k)}), \quad (3.12)$$

esta función obtiene altos valores cuando la distancia entre i y $c(q)$ es menor que la asignación actual i a $c(k)$ y la distancia entre j y $c(k)$ es menor que la asignación actual j a $c(q)$. Entonces, se aplica la configuración de movimiento con la mejor evaluación de esta función.

Los dos movimientos se ilustran en la Figura 3.2. La Figura 3.2(a) representa el movimiento de inserción. En estos ejemplos se asume que la capacidad residual de cada centro es $r(k) = 26$ y $r(q) = 18$, y la demanda de cada nodo es $w_i = 5$ y $w_j = 8$, donde i indica el nodo «cuello de botella» y j es designado como el nodo más atractivo para movimiento. La Figura 3.2(a) representa el movimiento de inserción. Aquí, el nodo i cambia su localización desde k a q reduciendo el costo actual de la solución a doce. La Figura 3.2(b) representa el movimiento de intercambio, donde los nodos i y j son intercambiados, reduciendo el costo de la solución a doce.

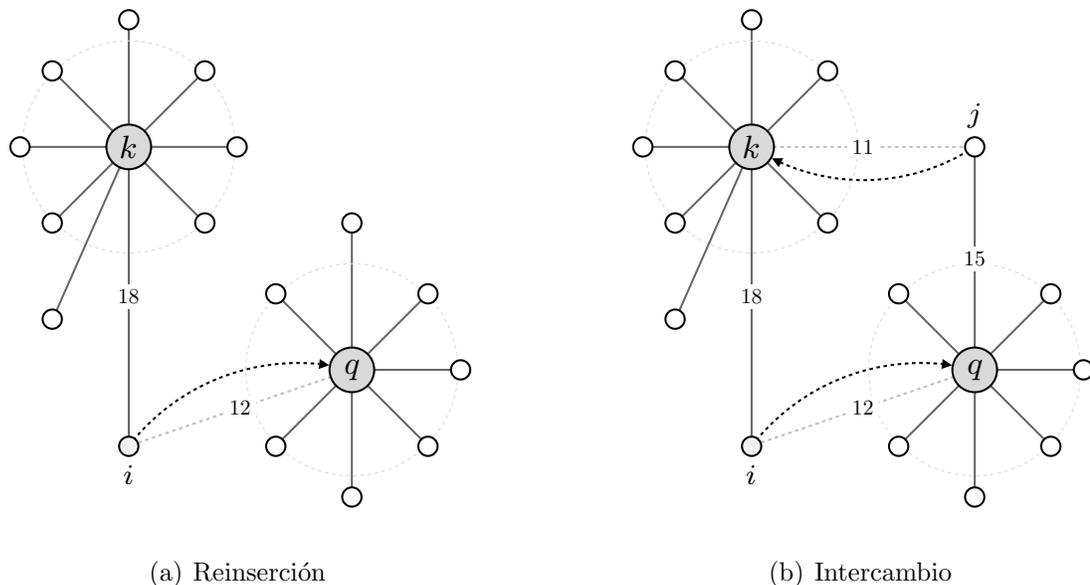


Figura 3.2: Ilustración de las estructuras de vecindario en el método VND.

PROCEDIMIENTO DE AGITACIÓN

Los métodos de perturbación y VND descritos anteriormente consideran movimientos donde los centros no tienden a cambiar de manera significativa, lo que ocasiona que en algún punto del trayecto de la búsqueda la estructura de la solución no sea modificada y se estanque en un óptimo local. El esfuerzo computacional de cada evaluación del movimiento es razonable.

Sin embargo, es posible forzar un mecanismo agresivo de destrucción/reconstrucción que propiciaría el cambio de la estructura parcial de la solución. Esto se logra a través de un procedimiento definido como «Agitar» (*shake* en inglés) que se ejecuta cuando se encuentra un óptimo local después de aplicar iteradamente las fases de perturbación y VND. Por supuesto, este mecanismo requiere un esfuerzo computacional mayor, pero, como se mostrará en la evaluación computacional, aporta significativamente en la reducción del objetivo. El método de agitar realiza una eliminación y reconstrucción total de varias particiones, lo cual diversifica en gran medida el trayecto de la búsqueda. El proceso, definido en el Algoritmo 5 funciona de la siguiente manera. Dada una p -partición X y un nodo «cuello de botella» $i \in \mathcal{J}(X)$,

Algoritmo 5 Agitación de la p -partición X **Require:** p -partición X

```

1:  $q = \lceil \ln(p + 1) \rceil$ 
2: for all  $i \in \mathcal{J}(X) : k_1 = \ell(i)$  do
3:    $W \leftarrow \{X_{k_1}, \dots, X_{k_q}\}$ 
4:    $X \leftarrow X \setminus W$ 
5:    $X' \leftarrow \text{Construccion}(W, q)$ 
6:    $X \leftarrow X \cup X'$ 
7: end for
8: return  $X$ 

```

se define el conjunto $W = \{X_{k_1}, \dots, X_{k_q}\}$, de q particiones con $\{k_1, \dots, k_q\} \subset K$, que corresponde a los índices de los q centros cercanos a i . Este es un conjunto ordenado, es decir, $c(k_1)$ es el centro más cercano a i . Es notable observar que k_1 no necesariamente es igual a $\ell(i)$, es decir, es posible que el nodo i no se encuentra actualmente asignado a su centro más cercano. Para cada nodo $i \in \mathcal{J}(X)$, tal que $k_1 = \ell(i)$, las q particiones cercanas a i son destruidas, esto es cada nodo $j \in W$ es desconectado de su centro actual. La razón de esto es que tiene más sentido destruir a las particiones donde los nodos de «cuello de botella» se asignan a su centro más cercano, ya que es más probable que cambie realmente el valor de la función objetivo minmax. Con la finalidad de mantener una relación entre el número de centros destruido y p , se ha fijado $q = \lceil \ln(p + 1) \rceil$. Entonces la eliminación de las particiones de la solución actual es $X \leftarrow X \setminus W$. Ahora, usando el método de construcción definido en 3.1.1, se construye una nueva solución parcial X' para reasignar los nodos con un número de instalaciones requeridas q . Finalmente $X \leftarrow X \cup X'$ es la nueva solución actual.

Es fácil notar que durante la reconstrucción se introducen mecanismos de aleatoriedad, influenciados por la fase de construcción. El mecanismo de agitar la solución no garantiza la reducción de nodos «cuello de botella» o una mejora en el valor de la función objetivo, sin embargo, la destrucción agresiva diversifica parcialmente la estructura de la solución actual, seleccionando otro conjunto de centros potenciales lo cual puede ser deseable para guiar el método a otras regiones de la búsqueda más prometedoras durante la fase de mejora a ser realizada posteriormente.

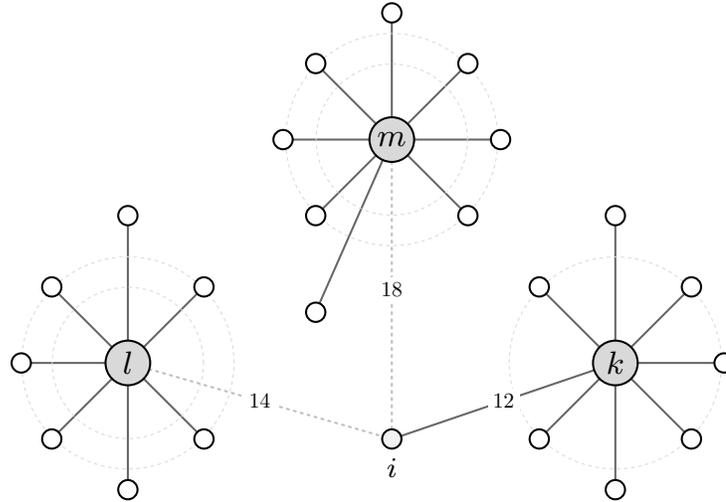


Figura 3.3: Segmento de una solución que es elegible para aplicar el método de agitar.

La Figura 3.3 muestra un segmento de solución que es elegible para aplicar el método de agitar. En este ejemplo, i es un nodo «cuello de botella» actualmente asignado al centro k bajo un criterio de capacidad y distancia dado por la función (3.4), el cual coincide con el criterio de distancia, es decir, $k = \ell(i)$. Por lo tanto, bajo el criterio previamente definido y con $q = 3$, la región formada por las particiones con centro en k, l y m es destruida. Entonces el segmento es reconstruido usando la fase de construcción definida en el Algoritmo 2, logrando una nueva solución.

3.2 ANÁLISIS DE LA COMPLEJIDAD DEL ALGORITMO

Se realiza un análisis algorítmico de los principales componentes del método propuesto, bajo el supuesto del peor caso posible al cual se pueda enfrentar y en consecuencia obteniendo una cota superior asintótica $\mathcal{O}(f(n))$. El análisis asintótico pone de manifiesto como aumenta en el límite el tiempo de ejecución en función del aumento del tamaño de la entrada. Con el fin de facilitar la reducción, establecemos las siguientes variables dentro del análisis. Sea n el número de nodos en el conjunto V , p el número de instalaciones ubicar, t el número de iteraciones que realizar el algoritmo, es decir $r_{\text{máx}}$ y $m = n - p = |V \setminus P|$.

- *Construcción:* Se asume que $p = 1$; entonces, el número de operaciones requeridos para calcular la probabilidad definida en (3.3) es de $\mathcal{O}(m)$, se concluye que la complejidad algorítmica de la etapa (a) es $\mathcal{O}(mp)$. En la siguiente etapa, cada usuario es asignado a un centro que minimize la función $\phi(j, k)$ definida en (3.4) misma que tiene un orden de operaciones de $\mathcal{O}(np)$. Actualizar el centro requiere $\mathcal{O}(n^2)$ operaciones y por lo tanto la etapa (b) tiene una complejidad de $\mathcal{O}(mn^2)$. Dado que la complejidad de la etapa (b) es mayor que (a), se concluye que la complejidad total del procedimiento de construcción definido en el Algoritmo 2 es de $\mathcal{O}(mn^2)$.
- *Perturbación:* Se asume que $p = 1, \alpha = 1.0$. Durante la etapa de destrucción es necesario calcular la probabilidad (definida en (3.6)) de desconexión para cada nodo de una partición X_k , lo cual necesita un número de operaciones de $\mathcal{O}(n)$. Se observa que el número de nodos a desconectar en este caso $|W| = m$, por lo que se requiere $\mathcal{O}(mp)$ operaciones para realizar la inclusión de los elementos en W . En consecuencia la etapa de destrucción tiene una complejidad de $\mathcal{O}(np)$. La siguiente etapa reasigna cada usuario desconectado a un centro que minimize la función $\phi(j, k)$ definida en (3.4) misma que tiene un orden de operaciones de $\mathcal{O}(np)$, seguida de una actualización de los centro lo cual necesita $\mathcal{O}(n^2p)$ operaciones. La etapa de reconstrucción tiene un mayor costo que la destrucción, se concluye entonces que la complejidad de la perturbación definido en el Algoritmo 3 es $\mathcal{O}(n^2p)$.
- *VND:* En este componente, la deducción de la complejidad de forma analítica no resulta sencilla. Se observa que el algoritmo es no determinista por no definir un número fijo de pasos para concretar el procedimiento, dado que h puede ser reiniciada a 1 cada vez que ocurre una mejora en el valor de la función objetivo. Sin embargo es posible modelar la probabilidad de reinicio de h a través de un proceso estocástico basado en la demostración realizada por Qu, Xu y Kendall [72] para un algoritmo VND que resuelve un problema de ruteo con restricciones de retraso.

Demostración: *Probar la transición de i a j en un árbol de expansión.* Para una red de v nodos hay v^{v-2} posibles árboles de expansión, por lo cual el total de árboles de Steiner es también por v^{v-2} . Entonces, se considera una cadena de Márkov [59] de v^{v-2} estados, donde cada estado corresponde a un árbol de expansión. Se ordenan los estados en un orden descendente con respecto al costo del árbol de Steiner seguido de reemplazar cada estado de la lista ordenada con v copias de si mismo, lo cual da un total v^{v-1} . En la cadena de Márkov, la arista de transición de un estado i va solo hacia el estado siguiente j de i . Se asume que cada posible transición es igualmente probable. Entonces, el problema de transición de i a j es

$$p_{ij} = \frac{1}{i-1} (1 \leq j \leq i, P_{11} = 1). \quad (3.13)$$

Sea q_i el número de transiciones necesarias para ir del estado i a 1, el valor esperado es $E[q_i] = \log(i)$. Por lo tanto si el algoritmo inicia desde el estado con mayor costo, es decir v^{v-1} , el número esperado de transiciones $\mathcal{O}(\log(v^{v-1})) = \mathcal{O}(v \log(v))$. ■

Por lo anterior, el número esperado de iteraciones de las estructuras de vecindario es $\mathcal{O}(v \log(v))$. Aquí se tienen dos vecindarios \mathcal{N}_1 y \mathcal{N}_2 , con una complejidad $\mathcal{O}(mp)$ y $\mathcal{O}(m^2)$ respectivamente; después de la selección de movimiento se realiza una actualización de centros de la solución modificada X' que necesita $\mathcal{O}(n^2p)$ operaciones, ésta es la operación con mayor costo dentro del ciclo. Por lo tanto, la complejidad de Algoritmo 4 es

$$\mathcal{O}(v \log(v)n^2p). \quad (3.14)$$

- *Agitación:* Se asume que $p = 1$, entonces $q = 1$, por lo que la totalidad de la solución es destruida y reconstruida, la complejidad de este algoritmo esta dominada por la complejidad del procedimiento de construcción el cual se demostró ser de $\mathcal{O}(mn^2)$, en conclusión la complejidad del procedimiento de agitación definido en el Algoritmo 5 es también $\mathcal{O}(mn^2)$.

Finalmente, la complejidad del procedimiento principal IGLS/VND definido en el Algoritmo 1 está dominada por ciclo de la búsqueda local (pasos 4–11) con t número de iteraciones. Dentro del ciclo, el componente con mayor complejidad es la perturbación con $\mathcal{O}(n^2p)$, por lo tanto se concluye que la complejidad total del procedimiento propuesto es

$$\mathcal{O}(tn^2p). \tag{3.15}$$

CAPÍTULO 4

MÉTODO DE SOLUCIÓN EXACTO

En esta sección se define la implementación computacional de un método de ramificación y generación de columnas con planos cortantes para el $CpCP$. Los aspectos teóricos, notación y métodos sobre descomposición son descritos en el Apéndice D. El marco del método de ramificación y generación de columnas (B&P, por sus siglas en inglés, *branch-and-price*) ha sido ampliamente utilizado en años recientes para la solución de problemas de optimización combinatoria, Desrosiers y Lübbecke [17, 18] realizan una amplia revisión de la literatura de este método.

La integración de B&P con el método de planos cortantes es denominada como ramificación y generación de columnas con planos cortantes (BP&C, por sus siglas en inglés: *branch-price-and-cut*). Este método ha sido estudiado por diversos autores: Savelsbergh [84], Barnhart et al. [8], Barnhart, Hane y Vance [7], Vanderbeck [89] y más recientemente Poggi de Aragão y Uchoa [16], Desrosiers y Lübbecke [18] y Gamrath [33]. La Sección B.2.4 define los principios y funcionamiento del método.

En particular, en el área de localización, entre los trabajos con mayor relevancia aplicando esta técnica encontramos el de Díaz y Fernandez [20] para el problema capacitado de localización de instalaciones con fuente única (SSCPLP, por sus siglas en inglés: *single source capacitated plant location problem*); Senne et al. [86], quienes definen un algoritmo para el pMP ; Ceselli y Righini [13], quienes proponen el método para el $CpMP$ y más recientemente Klose y Görtz [51], desarrollan un modelo de solución para el problema de localización de instalaciones capacitado (CFLP, por sus siglas en inglés: *capacitated facility location problem*).

Para esta implementación se ha usado el marco de trabajo para métodos de descomposición DIP [31] (por sus siglas en inglés: *decomposition in integer programming*) con una interfaz en Python llamada Dippy creada por O’Sullivan et al. [68] y que actualmente se encuentra ya integrada al marco de COIN-OR (por sus siglas en inglés: *Computational Infrastructure for Operations Research*). Dippy provee una poderosa interfaz que invocan a las funciones de DIP, pero sin sacrificar usabilidad y flexibilidad. La esencia del software DIP es explicada con mayor claridad en la Sección B.3.

4.1 MÉTODO DE SOLUCIÓN PROPUESTO

El método propuesto asume la existencia de cotas de buena calidad para el CpCP, con la finalidad de reducir el número de variables del modelo y propiciar una solución eficiente por parte de DIP. Se calculan las cotas usando la primera fase del método de Albareda-Sambola, Díaz y Fernández [4]. Entonces, dada una cota inferior (LB) y superior (UB), fijamos primeramente el dominio de la variable de linealización z como $LB \leq z \leq UB$, tal que $LB, UB \in \mathbb{N}$ y las variables de decisión $x_{ij} = 0$ si la distancia $d_{ij} > UB$ de tal manera que eliminamos asignaciones infactibles; por nomenclatura definimos a $\delta = UB$ entonces el modelo matemático que incluye estas restricciones es el siguiente

$$(CpCP) v(\delta) = \text{Minimizar } z \tag{4.1}$$

$$\text{s. a. } \sum_{i \in V: d_{ij} \leq \delta} d_{ij} x_{ij} \leq z \quad j \in V, \tag{4.2}$$

$$\sum_{i \in V: d_{ij} \leq \delta} x_{ij} = 1 \quad j \in V, \tag{4.3}$$

$$\sum_{j \in V: d_{ij} \leq \delta} w_j x_{ij} \leq s_i y_i \quad i \in V, \tag{4.4}$$

$$\sum_{i \in V} y_i = p, \tag{4.5}$$

$$x_{ij}, y_i \in \{0, 1\} \quad i \in V, j \in V. \tag{4.6}$$

En DIP, una vez que el modelo ha sido descrito, se tiene la opción de definir o anular los diferentes componentes que afectarán el procesamiento del método, cualquier reformulación requerida se mantiene y realiza internamente y se interactúa sólo con la formulación original, además se proporciona una interfaz a través de librerías para la definición de los componentes específicos que se personalizan basado en la características del problema. DIP integra cada uno de estos componentes en un marco genérico por lo cual se facilita la implementación y diseño de un método al enfocar el esfuerzo sólo en la definición de los mecanismo esenciales. En este caso se definen los componentes de ramificación, generación de cortes, generación de columnas y columnas iniciales, que son descritos a continuación.

4.1.1 RAMIFICACIÓN

Se definen dos estrategias de ramificación accediendo a las variables de la formulación original. La primera estrategia es común de encontrar en las implementaciones de algoritmos híbridos sobre B&B, en este caso la estrategia es similar a una de las usadas por Savelsbergh [84] para su B&P de un problema de asignación generalizada (GAP, por sus siglas en inglés: *generalized assignment problem*), la cual se basa en el valor fraccionario de las variables de decisión. Sea $\hat{x}_{ij} \notin \mathbb{Z}$ la variable de decisión con el mayor valor fraccionario, donde $\hat{x} \in \mathbb{R}^n$ es la solución LP de la relajación lineal, entonces la siguiente restricción de ramificación es agregada

$$x_{ij} \leq \lfloor \hat{x}_{ij} \rfloor, x_{ij} \geq \lceil \hat{x}_{ij} \rceil \quad (4.7)$$

La segunda estrategia utiliza las variables de localización y de manera similar. Fijar estas variables tienen un mayor efecto que fijar variables x , cuando una variable de localización y es fijada a 0, varias asignaciones pueden ser también fijadas a 0. Por lo tanto, resulta más efectivo fijar las variables y durante las primeras fases de exploración en el árbol de búsqueda. Sea $\hat{y}_i \notin \mathbb{Z}$ la variable de localización con el mayor fraccionario, donde $\hat{y} \in \mathbb{R}^n$ es la solución LP de la relajación lineal, entonces

la siguiente restricción de ramificación es agregada

$$y_i \leq \lfloor \hat{y}_i \rfloor, y_i \geq \lceil \hat{y}_i \rceil \quad (4.8)$$

Estas dos estrategias se combinan para formar un marco de ramificación. Si la segunda estrategia no logra formular una restricción de ramificación, es decir que no se encontró al menos un variable de localización \hat{y} con costo fraccionario o que toda variable $\hat{y}_i \in \mathbb{Z}$, entonces la primer estrategia basada en las variables \hat{x}_{ij} es aplicada. La naturaleza de la función objetivo minimax, y por lo tanto la asimetría del modelo, imposibilita la implementación de una estrategia de ramificación con mayor eficiencia que reduzca sustancialmente el árbol y espacio de soluciones del problema.

4.1.2 GENERACIÓN DE CORTES

En este componente se define el procedimiento para la generación de desigualdades válidas del modelo. DIP tiene una gran variedad de generadores comunes de desigualdades válidas integradas de forma genérica proporcionadas por COIN/CGL (por sus siglas en inglés: *cut generator library*). Sin embargo en ocasiones es preferible definir un método que explote las características del problema y genere desigualdades válidas que mejoren el rendimiento.

Marchand y Wolsey [58], realizan un compendio de varios tipos de desigualdades efectivas para problemas ILP. Uno de estas son las desigualdades de peso que toman en cuenta las restricciones de capacidad del problema. Este estrategia fue implementada por Martin y Weismantel [61] para el problema de la mochila con diferentes características (KP, por sus siglas en inglés: *knapsack problem*) y más recientemente O’Sullivan et al. [68], quienes las utilizan en la generación de cortes de un BP&C para un problema de localización de instalaciones capacitado (CFLP, por sus siglas en inglés: *capacitated facility location problem*).

Para cada instalación potencial $i \in V$, se tiene un subconjunto $\mathcal{S}_i \subseteq \{1, \dots, n\}$ formado de una manera voraz a partir de la mejor asignación encontrada dependiendo de los valores fraccionarios de las variables de decisión x_{ij} , donde $x \in \mathbb{R}^n$, tal que $\sum_{j \in \mathcal{S}_i} w_j x_{ij} \leq s_i$; en este caso la inclusión de un nodo j a \mathcal{S}_i es limitada por la restricción de la cota superior δ , es decir $d_{ij} < \delta$. Además se tiene un vector μ_i definido por

$$\mu_i = s_i - \sum_{j \in \mathcal{S}_i} w_j x_{ij}. \quad (4.9)$$

El Algoritmo 6 define la creación del conjunto \mathcal{S}_i y el vector μ_i .

Algoritmo 6 Creación del conjunto \mathcal{S}_i y vector μ_i

Require: $i \in V = \{1, \dots, n\}, \delta \in \mathbb{N}$

- 1: $\mu_i = s_i, i \in V$
 - 2: **repeat**
 - 3: $Q \leftarrow (i, j) \in V \times V : j \notin \mathcal{S}_i, w_j \leq \mu_i, d_{ij} < \delta$
 - 4: **if** $Q \neq \emptyset$ **then**
 - 5: $(\hat{i}, \hat{j}) \leftarrow \arg \max_{(i,j) \in Q} x_{ij}$
 - 6: $\mu_i \leftarrow -w_{\hat{j}}$
 - 7: $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{\hat{j}\}$
 - 8: **end if**
 - 9: **until** $Q = \emptyset$
 - 10: **return** \mathcal{S}_i
-

Se genera la siguiente desigualdad válida para el problema, la cual es agregada al problema maestro restringido.

$$\sum_{j \in \mathcal{S}_i} w_j x_{ij} + \sum_{j \in V : j \notin \mathcal{S}_i, d_{ij} < \delta} (w_j - \mu_i)^+ x_{ij} \leq s_i - \mu_i. \quad (4.10)$$

4.1.3 GENERACIÓN DE COLUMNAS

En este apartado se definirán los elementos para la generación de columnas. Con este método se describe un algoritmo para la resolución de cada uno de los subproblemas. Primeramente es necesario redefinir el modelo CpCP en términos de descomposición de Dantzig-Wolfe, que denota el *problema maestro* (PM). DIP sólo interactúa con la formulación original, pero es necesario realizar esta reformulación para definir el *problema de precio*.

PROBLEMA MAESTRO

Sea $K^i = \{1, \dots, k\}$ el conjunto de índices de todas las asignaciones factibles para la instalación i dado por $A^i = \{X_1^i, \dots, X_k^i\}$. Sea $X_k^i = \{a_{1k}^i, \dots, a_{nk}^i\}$ un patrón de asignación factible sólo si $\sum_{j \in V} w_j a_{jk}^i \leq s_i$ y $d_{ij} a_{jk}^i \leq \delta$, donde

$$a_{jk}^i = \begin{cases} 1, & \text{si el usuario } j \text{ está asignado a la instalación } i \text{ en el } k\text{-ésimo patrón de } i, \\ 0, & \text{otro caso;} \end{cases}$$

además se define la variable de decisión

$$y_k^i = \begin{cases} 1, & \text{Si la } k\text{-ésima asignación factible es seleccionada para la instalación } i, \\ 0, & \text{otro caso;} \end{cases}$$

entonces, el problema puede ser reformulado de la manera siguiente acorde a la descomposición de Dantzig-Wolfe.

$$(SCpCP) \ v(\delta) = \text{Minimizar } z \tag{4.11}$$

$$\text{s. a. } \sum_{i \in V} \sum_{k \in K^i} (d_{ij} a_{jk}^i) y_k^i \leq z, \quad j \in V, \tag{4.12}$$

$$\sum_{i \in V} \sum_{k \in K^i} a_{jk}^i y_k^i = 1, \quad j \in V, \tag{4.13}$$

$$\sum_{i \in V} \sum_{k \in K^i} y_k^i = p, \tag{4.14}$$

$$\sum_{k \in K^i} y_k^i \leq 1, \quad i \in V, \tag{4.15}$$

$$y_k^i \in \{0, 1\} \quad k \in K^i, i \in V. \tag{4.16}$$

Cada columna del modelo corresponde a una asignación factible que satisface las restricciones de capacidad de una instalación potencial. (4.11) y (4.12) representan la linealización de la función objetivo original $\min_{i \in V} \max_{j \in V} d_{ij} x_{ij}$ en términos de descomposición. Las restricciones (4.13) garantizan que cada nodo es asignado a un único centro i . Las restricciones (4.14) indican que el total de asignaciones facti-

bles es a lo más p ; las restricciones (4.15) aseguran que sólo una solución factible es asociada al mismo centro i .

Cada K^i de conjuntos factibles, contiene un número exponencial de elementos. Debido a que PM no puede resolverse directamente por el número exponencial de sus columnas, un *problema maestro restringido* (PMR) es definido. PMR tiene un número relativamente menor de columnas, el cual puede ser resuelto a optimalidad. La búsqueda de estas columnas se basa en que su costo reducido sea negativo, producto de resolver un *problema de precio*. Cuando ya no existan columnas con costo reducido negativo, la solución óptima del PMR es también el óptimo del PM, entonces este valor es una cota dual válida para usarse dentro del B&B.

PROBLEMA DE PRECIO

Este componente es una de las partes más importantes dentro de un método de BP&C, pues permite delimitar el número de columnas que conformar el nuevo PMR. Basados en el PM se realiza la formulación del *problema de precio* (PP). Al modelo SCpCP o problema maestro restringido se le aplica una relajación Lagrangiana en las restricciones (4.12), (4.13), (4.14) y (4.15); la función dual resultante se muestra a continuación:

$$\begin{aligned} \text{LSCpCP}(\mu, \lambda, \eta, \rho) &= \sum_{j \in V} (z\mu_j + \lambda_j) + \eta p + \sum_{i \in V} \rho_i + \\ \text{Min } z &- \sum_{i \in V} \sum_{k \in K^i} \left[\sum_{j \in V} (d_{ij}u_j - \lambda_j)a_{jk}^i - (\eta - \rho_i) \right] y_k^i, \end{aligned} \quad (4.17)$$

donde $\mu \in \mathbb{R}_+^n$, $\lambda \in \mathbb{R}^n$, $\eta \in \mathbb{R}$ y $\rho \in \mathbb{R}_+^n$ son los vectores de las variables duales no negativos correspondientes a las restricciones (4.12), (4.13), (4.14) y (4.15) respectivamente. Entonces, el costo reducido de una columna $k \in K^i$ es

$$r(\mu, \lambda, \eta, \rho) = z - \sum_{j \in V} (d_{ij}u_j - \lambda_j)a_{jk}^i - \eta + \rho_i. \quad (4.18)$$

Localizar las columnas con el costo reducido negativo que formara en el nuevo PMR implica resolver un PP por cada instalación potencial $i \in V$, esto se modela de la manera siguiente

$$(PP_i) u = \text{Min } z - \sum_{j \in V} (d_{ij}u_j - \lambda_j)a_{jk}^i - \eta + \rho_i \quad (4.19)$$

$$\text{s. a. } \sum_{j \in V} w_j a_{jk}^i \leq s_i, \quad (4.20)$$

$$a_{jk}^i \in \{0, 1\}, j \in V. \quad (4.21)$$

Se observa, que dentro de esta formulación se encuentra implícito un KP binario para cada $i \in V$, el cual se desagrega como

$$(KP_i) w = \text{Max } \sum_{j \in V} (\lambda_j - d_{ij}u_j)a_{jk}^i \quad (4.22)$$

$$\text{s. a. } \sum_{j \in V} w_j a_{jk}^i \leq s_i, \quad (4.23)$$

$$a_{jk}^i \in \{0, 1\}, j \in V, \quad (4.24)$$

por lo cual PP_i puede ser redefinido de la forma siguiente

$$(PP_i) u = \text{Min } z + KP_i - \eta + \rho_i. \quad (4.25)$$

Este es finalmente el *problema de precio*. La complejidad de KP_i es NP-duro, pero ya que no es necesario encontrar la columna con el costo reducido más negativo, métodos de búsqueda heurísticos pueden ser utilizados. Se conoce la existencia de eficaces algoritmos pseudo-polinomiales e implementaciones para el KP binario; por ejemplo, O'Sullivan et al. [68] implementan para este paso una solución a través de programación dinámica logrando aumentar el rendimiento de su método, el cual es utilizado también en esta implementación.

GENERACIÓN DE COLUMNAS INICIALES

Una de las opciones de DIP permite generar genéricamente un conjunto de columnas iniciales basadas en la formulación original. Sin embargo, en ocasiones es preferible personalizar este componente de tal manera que tome en cuenta las características del modelo para lograr una mayor calidad de las columnas iniciales agregadas al PMR, propiciando una convergencia más rápida del algoritmo. Para esta implementación se ha decidido generar un conjunto de columnas factibles para cada $i \in V$ a través de un sencillo método voraz. El procedimiento es mostrado en el Algoritmo 7.

Algoritmo 7 Generación de columnas iniciales

Require: $V = \{1, \dots, n\}$ y $\delta \in \mathbb{N}$

```

1:  $X \leftarrow \emptyset$ 
2: for all  $i \in V$  do
3:    $U \leftarrow V$ 
4:   while  $U \neq \emptyset$  do
5:      $\hat{j} \leftarrow \arg \min_{j' \in U} \{d_{ij'} : d_{ij'} < \delta\}$ 
6:     if  $\sum_{j' \in X_i} w_{j'} + w_{\hat{j}} \leq s_i$  then
7:        $X_i \leftarrow X_i \cup \{\hat{j}\}$ 
8:        $U \leftarrow U \setminus \{\hat{j}\}$ 
9:     else
10:      break
11:    end if
12:  end while
13: end for
14: return  $X$ 

```

Para cada i se agregan nodos a una partición X_i , bajo el criterio de la menor distancia respecto a i (paso 5), pero sin violar la cota superior δ , la partición es completada hasta que la capacidad del nodo i se ha agotado o el conjunto de nodos candidatos es vacío. Por las restricciones de capacidad y de distancia, el método siempre asegura la generación de una columna o partición factible para el nodo i . El método finaliza cuando se ha completado una asignación factible para cada nodo, retornando una n -partición, que contiene el conjunto de columnas válidas que son agregadas inicialmente al problema maestro restringido.

CAPÍTULO 5

ESTUDIO COMPUTACIONAL

En este capítulo se reporta un estudio computacional que muestra el rendimiento de los métodos propuestos para la solución del $CpCP$, además de una completa comparación frente a los métodos encontrados en literatura. Se definen inicialmente las condiciones experimentales a las que está sujeto este estudio así como el conjunto y estructura de las instancias de prueba que se utilizan. Esta evaluación empírica se respalda con su correspondiente análisis estadístico basado en un adecuado diseño experimental [64].

5.1 OBJETIVOS

Este estudio computacional tiene como propósito principal probar la eficiencia de los métodos para el vasto y diverso conjunto de instancias presente en la literatura cuya estructura incluye los casos generales y extremos a los cuales cada algoritmo puede ser sometido. Además, otro de los objetivos generales es realizar una comparación entre todos los métodos encontrados en la literatura. A continuación se definen los objetivos específicos.

- Captar mediante un estudio estadístico, los parámetros del heurístico que ayuden a obtener soluciones de mejor calidad; en este caso el parámetro a ajustar es α .
- Evaluar el desempeño del método heurístico desarrollado y cada uno de sus componentes con base en un diseño experimental adecuado.

- Realizar una comparación entre la heurística y todos los métodos encontrados en la literatura y argumentar los resultados.
- Evaluar el desempeño del método exacto desarrollado con base en un diseño experimental adecuado.

5.2 CONDICIONES EXPERIMENTALES

El método heurístico propuesto ha sido codificado en C++, así mismo la heurística de Scaparra, Pallottino y Scutellà [85] (SPS). Los métodos exactos de Özsoy y Pinar [69] (OP) y Albareda-Sambola, Díaz y Fernández [4] (ADF) se encuentran codificados en ANSI C, en este caso ILOG CPLEX 12.5 es usado como optimizador de MILP. El método exacto propuesto ha sido codificado la interfaz en Python para DIP. Se establece para todos los métodos exactos un límite de tiempo de 1 hora y un uso de memoria de 4 GB como criterio para finalizar la ejecución en los conjuntos de instancias A, B, C, D-* y 6 horas y 4 GB para el conjunto E. Todos los métodos en ANSI C/C++ se han compilado usando gcc/g++ versión 4.4.5 bajo Debian 6.0.8 GNU/Linux Kernel 2.6.32-5, 64 bits. Para cada uno de los experimentos se utilizó un *cluster* AMD Opteron con 2.0 GHz (x16) con 32 GB en RAM.

5.2.1 INSTANCIAS DE PRUEBA

En los experimentos se usan cuatro diferentes conjuntos de instancias. Para el CpCP no se han formulado instancias específicas de prueba en la literatura; sin embargo se prueban y comparan todos los métodos con instancias generadas para otros problemas de localización y usadas previamente en trabajos de este problema. A continuación se define cada conjunto de instancias de prueba.

- (A) Biblioteca-OR: Este conjunto de instancias, propuesto en [11] para el CpMP, contienen dos grupos de 10 instancias con la misma capacidad de instalaciones, el primero tiene 50 nodos de demanda y 5 instalaciones a ser ubicadas, el segundo tiene 100 nodos de demanda y 10 instalaciones a ser ubicadas.

- (B) Galvão y ReVelle: Este conjunto de instancias fue generado por Scaparra, Pallottino y Scutellà [85] específicamente para el $CpCP$ basado en el conjunto de instancias de Galvão and ReVelle para el problema de localización de máxima cobertura [32]. El conjunto contiene instancias con 100 y 150 usuarios y 5 a 15 centros, con capacidades de instalación variables. El conjunto original está compuesto por 2 redes creadas aleatoriamente.
- (C) Lorena y Senne: Este conjunto, propuesto en [56] para el $CpMP$, incluye 6 instancias cuyo rango de nodos es de 100 a 402 usuarios y de 10 a 40 centros a ser ubicados.
- (D) Ceselli y Righini: Éste es un reciente conjunto de instancias añadido a la Librería-OR propuesta en [13]. Este se considera como un conjunto difícil de resolver para los problemas de localización capacitados como el $CpMP$ y el $CpCP$, lo cual puede estar influenciado significativamente por la relación entre n y p . Ese conjunto es compuesto por 4 subconjuntos denotados por α, β, γ y δ , cada subconjunto consiste de 40 instancias, formadas a partir de un conjunto inicial con las siguientes características: 20 de ellas son grafos de 50 y 100 nodos y $p = \frac{n}{10}$; las otras 20 instancias fueron generadas aleatoriamente con una cardinalidad de 150 (15 centros) y 200 (20 centros), además todas las capacidades son fijadas en 120 unidades. Por consiguiente, las mismas 40 instancias se configuraron con diferentes valores de p y s_i en cada diferente subconjunto: α con $p = \lfloor \frac{n}{10} \rfloor$, β con $p = \lfloor \frac{n}{4} \rfloor$, γ con $p = \lfloor \frac{n}{3} \rfloor$ y δ con $p = \lfloor \frac{2n}{5} \rfloor$. El total de la capacidad fue definida para todos los subconjuntos por la configuración $s_i = \lceil 12 \frac{n}{p} \rceil$. Por lo tanto, se describen como cuatro conjuntos de datos diferentes, es decir D- α , D- β , D- γ , y D- δ , cuya literal expresa la complejidad de cada subconjunto de instancias, siendo D- α el conjunto fácil y D- δ el complejo.
- (E) Reinelt: Éste conjunto fue generado por Lorena y Senne [56] específicamente para el $CpMP$ basado en el conjunto de instancias de la Biblioteca TSPLIB, recopilada por Reinelt [77]. El conjunto contiene 5 instancias con 3038 usuarios y de 600 a 1000 centros a ser ubicados con una misma capacidad por instancia,

calculado por $s_i = \left\lceil \frac{\sum w_j}{p \times 0.8} \right\rceil$. Este es considerado un conjunto a gran escala por el número de nodos.

Los conjuntos de instancias de A, C, D-* y E no incluyen la matriz de distancias, sólo contemplan la ubicación geográfica (x, y) de cada nodo en V , por lo que se procede durante la lectura de la instancia a calcular la distancia euclídea entre cada par de nodos usando la función (A.1).

5.3 EVALUACIÓN DEL MÉTODO HEURÍSTICO

En esta sección se reporta la evaluación del método heurístico. Primeramente se procede a realizar la calibración de los parámetros del método a través de un análisis estadístico y posteriormente se ejecuta una extensa experimentación computacional del método propuesto, así como de los métodos exactos y heurísticos existentes. Finalmente se muestran, comparan y argumentan los resultados obtenidos, los cuales serán determinante para aportar razonables conclusiones del método.

Para definir el conjunto de criterios a evaluar, el presente estudio se ha basado en las consideraciones presentadas por Barr et al. [9], que consiste en medir la calidad de la solución, esfuerzo computacional y robustez del método. Se utilizan los métodos exactos de Özsoy y Pınar [69] y Albareda-Sambola, Díaz y Fernández [4] para calcular la solución exacta o, en su defecto, una cota inferior cuando no sea posible alcanzar optimalidad; lo anterior, con la finalidad de obtener una mejor estimación de la desviación relativa de la solución heurística respecto al óptimo.

5.3.1 CALIBRACIÓN DE PARÁMETROS

El propósito del primer experimento es afinar la heurística con respecto al parámetro de destrucción α para cada conjunto de instancias, lo cual se lleva a cabo mediante un detallado análisis estadístico. Con el fin de facilitar esta tarea, se ha procedido a desarrollar un *script* en el lenguaje R versión 3.0.2 [73]. La variable de respuesta estudiada es el promedio respecto al porcentaje de desviación relativa

(DR) de la manera siguiente

$$\text{DR} = \frac{f(X) - f^{lb}}{f^{lb}} \times 100, \quad (5.1)$$

donde $f(X)$ es el resultado del valor de la función objetivo encontrado para cada instancia en particular y f^{lb} es su mejor cota inferior conocida. Esta cota fue obtenida aplicando ambos métodos exactos [4, 69] y seleccionando el mejor valor para la cota. En algunos casos este valor computado resulta ser el valor óptimo de la solución. El número de iteraciones de la heurística fue limitado a 100 para cada valor de $\alpha \in \{0.0, 0.1, \dots, 1.0\}$ para los conjuntos de instancias A, B, C, D-* y 50 para E, realizando 5 réplicas para este experimento. Los resultados son separados en 8 bloques de acuerdo al número de conjuntos de instancias con sus subconjuntos. Se estudia un sólo factor α para cada bloque compuesto por 5 réplicas, donde DR es la variable de respuesta. Primeramente los datos son analizados usando una prueba no-paramétrica de Kruskal-Wallis [52] con un intervalo de confianza (IC) del 95 % y se establece las siguientes hipótesis estadísticas: $H_0 : \mu_1 = \mu_2 = \dots = \mu_n$ y $H_1 : \text{al menos un par de medias } \mu_i \neq \mu_j$, donde μ_i es la media de la i -ésima réplica dentro del bloque k . Dicho de otra forma, la hipótesis nula H_0 se expresa como:

Hipótesis nula 1 *La media de las n réplicas del k -ésimo bloque son todas iguales y vienen de la misma distribución, por lo tanto el parámetro α no afectan la variable de respuesta DR para el k -ésimo bloque.*

La Tabla 5.1 muestra el resultado del estudio, donde la columna χ^2 expresa el valor del estadístico Chi cuadrado y k los grados de libertad. Para cada conjunto de instancias se obtiene un valor del estadístico p menor al error de 0.05; por lo tanto se rechaza la hipótesis nula 1 y se concluye que α tiene un efecto significativo en la variable de respuesta para cada uno de los conjuntos. La Figura 5.1 muestra tres gráficas en función de la desviación porcentual para el conjunto de instancias A (el resto de las gráficas se muestran en el Apéndice D). La primera es una sencilla gráfica de cajas basada en la relación entre los niveles α y DR. Aquí se observa

Tabla 5.1: Resultado de la prueba de Kruskal-Wallis, respecto al nivel definido por α y la variable de respuesta DR.

Conjunto	χ^2	k	p
A	432.39	10	$< 2.2 \times 10^{-16}$
B	150.43	10	$< 2.2 \times 10^{-16}$
C	216.94	10	$< 2.2 \times 10^{-16}$
D- α	1007.80	10	$< 2.2 \times 10^{-16}$
D- β	803.46	10	$< 2.2 \times 10^{-16}$
D- γ	691.62	10	$< 2.2 \times 10^{-16}$
D- δ	220.28	10	$< 2.2 \times 10^{-16}$
E	119.66	10	$< 2.2 \times 10^{-16}$

el efecto que tiene α sobre la variable de respuesta. Si $\alpha = 0.0$, DR tiene valores inferiores que los obtenidos en $0.1 \leq \alpha \leq 0.9$. Para el resto de las gráficas sólo se considera analizar el rango $[0.1, 0.8]$. La segunda despliega una gráfica de cajas de los grupos con su IC para la media usando una prueba de Friedman [28]. Esta gráfica es interesante porque permite localizar visualmente los grupos con respecto a otros, definiendo la significancia con color gris en la barra. Finalmente, la tercera es una gráfica de promedios con una prueba de Tukey con Diferencia Significativa Honesta (HDS, por sus siglas en inglés: *honest significant difference*) [1] con un IC del 95%. Esta gráfica proporciona una visión exacta de en qué nivel α la media obtiene su valor más bajo respecto a la variable de respuesta.

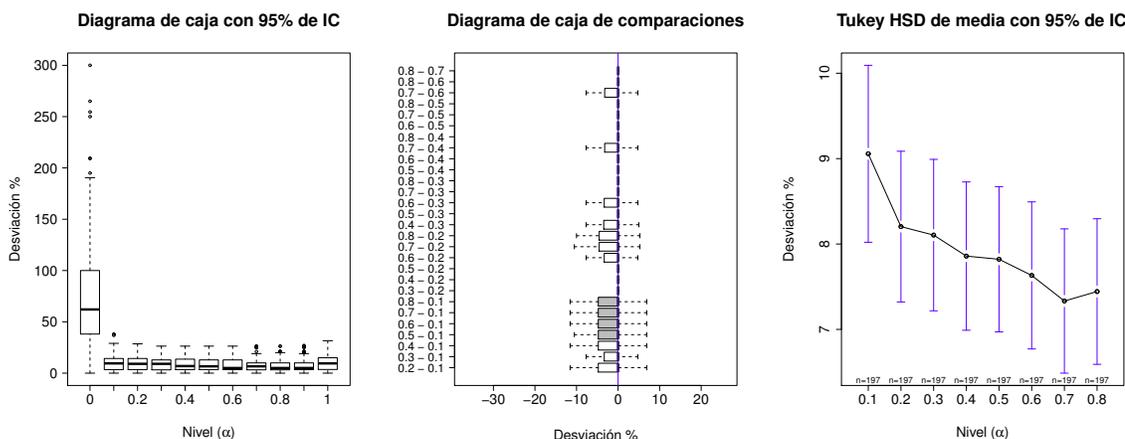


Figura 5.1: Análisis estadístico de la desviación porcentual en el conjunto de instancias A.

Basados en esta última prueba, la Tabla 5.2 muestra el conjunto de valores de α en los cuales se obtienen los valores más bajos de la variable de respuesta. Nótese cómo en alguna de las gráficas los niveles alcanzan valores similares respecto a la variable de respuesta, por lo tanto se ha seleccionado el menor valor de α entre estas similitudes con la finalidad de preservar el esfuerzo computacional.

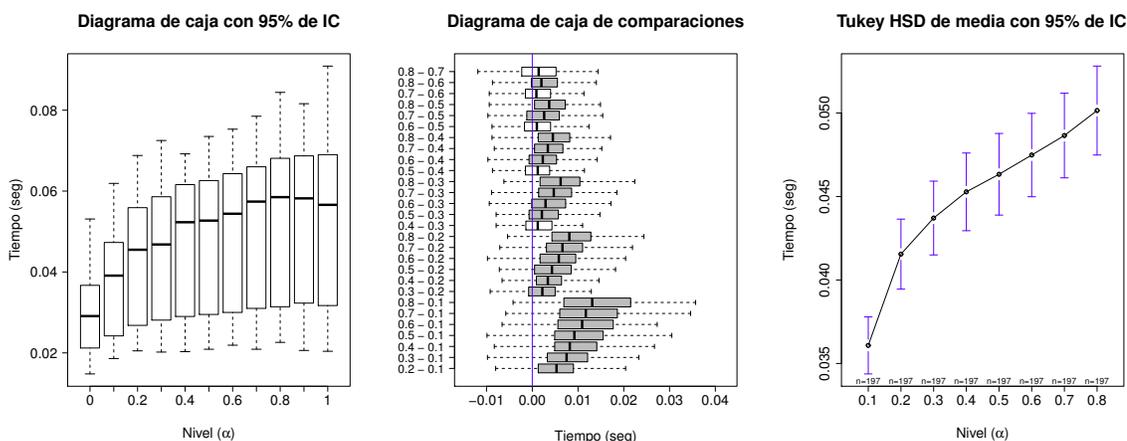


Figura 5.2: Análisis estadístico del tiempo en el conjunto de instancias A.

Cabe señalar que cuando α incrementa, también lo hace el nivel de destrucción de la solución y por lo tanto el esfuerzo computacional es mayor. La Figura 5.2 muestra las mismas gráficas de análisis pero ahora en función del tiempo respecto al nivel. Aquí se observa como el tiempo de cómputo se incrementa conforme α es mayor.

Tabla 5.2: Valores del parámetro α para cada conjunto de instancias, donde se obtienen los valores más bajos para la variable de respuesta DR.

Conjunto	α
A	0.7
B	0.6
C	0.7
D- α	0.6
D- β	0.4
D- γ	0.5
D- δ	0.4
E	0.4

5.3.2 COMPARACIÓN ENTRE HEURÍSTICAS

El principal propósito de este experimento es mostrar detalladamente una comparación entre el método heurístico propuesto (QR) y la única heurística existente en la literatura desarrollada por Scaparra, Pallottino y Scutellà [85] (SPS). Para el siguiente experimento se usan para cada conjunto de instancias los valores de α presentados en la Tabla 5.2 y se ejecuta la heurística QR usando 1000 iteraciones como límite para los conjuntos de instancias A, B, C y D-*, y 200 iteraciones para E, con 30 réplicas. Se ha decidido incluir los resultados obtenidos por los métodos de Özsoy y Pinar [69] (OP) y Albareda-Sambola, Díaz y Fernández [4] (ADF), por la razón de que se tiene el código fuente de los tres métodos existentes en la literatura y la heurística propuesta, ejecutando todos ellos en las mismas condiciones experimentales definidas en la Sección 5.2 lo que permite una comparación entre todos ellos en términos de esfuerzo computacional y calidad de la solución.

Las Tablas 5.3–5.10 presentan los resultados de la comparación entre los métodos para cada conjunto de instancias. En cada tabla, las dos primeras columnas representan el tamaño de la instancia medido por el número de nodos n y el número de particiones p . «Instancia» es el nombre de la instancia del problema y «Óptimo» indica el valor óptimo de la instancia o la mejor cota inferior conocida, denotada por «*» junto al valor, cuando no fue posible encontrar el óptimo con ninguno de los dos métodos exactos. La sección «Tiempo (s)» muestra por cada método el tiempo en segundos y «Desviación (%)» expresa el porcentaje de desviación relativa (DR) respecto al valor óptimo o mejor cota inferior conocida. En el caso de los métodos exactos, DR representa la desviación entre la mejor cota inferior y superior encontradas. Cabe señalar que para el método propuesto QR, se muestra el tiempo promedio utilizado en las 30 repeticiones independientes. Además «QR¹ %» y «QR² %» denotan el promedio y el mejor valor DR, respectivamente, entre todas las repeticiones. La Tabla 5.11 resume la comparación entre los métodos para todos los conjuntos de instancias en términos de su promedio de DR, tiempo de ejecución y número de soluciones infactibles. Una solución infactible, expresada por «-» es reportada si no fue

encontrada una solución factible después de que el criterio de paro fuera satisfecho. A continuación se discuten los resultados de cada una de las tablas.

Respecto al conjunto de instancias A (Tabla 5.3), los métodos exactos encuentran la solución eficientemente para el grupo instancias pequeñas (tamaño 50×5), teniendo un mejor desempeño que cualquier heurística. Sin embargo, para el grupo de mayor tamaño (tamaño 100×10), hay un par de instancias para las cuales es más difícil de resolver por los métodos exactos. En este conjunto el rendimiento de las heurísticas resultó ser más robusto que los métodos exactos en términos de tiempo computacional, ya que toma menos de 5.34 segundos en resolver cada una de las instancias. En términos de calidad de la solución, la heurística propuesta encuentra mejores soluciones en un tiempo menor que las reportadas por la heurística SPS.

Tabla 5.3: Comparación de los métodos en el conjunto de instancias A.

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
50	5	cpmp01	29	0.09	1.68	0.71	0.22	0.00	0.00	0.00	0.00	0.00
		cpmp02	33	0.93	3.53	0.97	0.20	0.00	0.00	0.00	0.00	0.00
		cpmp03	26	0.16	1.51	0.91	0.22	0.00	0.00	7.69	0.00	0.00
		cpmp04	32	0.81	2.60	1.01	0.21	0.00	0.00	0.00	0.00	0.00
		cpmp05	29	1.66	3.11	1.11	0.23	0.00	0.00	0.00	0.19	0.00
		cpmp06	31	2.75	3.82	1.06	0.22	0.00	0.00	3.23	3.07	0.00
		cpmp07	30	1.05	2.34	1.23	0.23	0.00	0.00	0.00	0.80	0.00
		cpmp08	31	0.88	2.89	1.09	0.24	0.00	0.00	0.00	1.29	0.00
		cpmp09	28	2.12	3.70	1.29	0.23	0.00	0.00	0.00	3.25	0.00
		cpmp10	32	3.42	4.63	2.50	0.23	0.00	0.00	9.38	12.81	0.00
		Promedio		1.39	2.98	1.19	0.22	0.00	0.00	2.03	2.14	0.00
100	10	cpmp11	19	11.68	3.99	9.34	0.58	0.00	0.00	5.26	9.62	0.00
		cpmp12	20	1.19	4.70	10.21	0.52	0.00	0.00	15.00	5.22	0.00
		cpmp13	20	8.83	4.16	9.32	0.54	0.00	0.00	10.00	0.64	0.00
		cpmp14	20	2.78	2.98	8.70	0.54	0.00	0.00	10.00	2.44	0.00
		cpmp15	21	7.01	5.46	9.64	0.56	0.00	0.00	9.52	3.73	0.00
		cpmp16	20	22.18	11.03	10.35	0.56	0.00	0.00	5.00	4.42	0.00
		cpmp17	22	97.34	551.76	9.91	0.57	0.00	0.00	4.55	4.70	4.55
		cpmp18	21	12.36	8.41	8.21	0.56	0.00	0.00	9.52	1.77	0.00
		cpmp19	21	6.43	12.85	10.16	0.58	0.00	0.00	9.52	5.66	0.00
		cpmp20	21	417.82	18.73	8.97	0.61	0.00	0.00	9.52	9.20	0.00
		Promedio		58.76	62.41	9.48	0.56	0.00	0.00	8.79	4.74	0.46
		Promedio Total		30.07	32.69	5.34	0.39	0.00	0.00	5.41	3.44	0.23

Analizando el conjunto de instancias B (Tabla 5.4) se observa que el método exacto toma considerablemente más tiempo que ambas heurísticas para encontrar una solución factible. En términos de la calidad en la solución de las heurísticas, QR obtiene ligeramente mejores soluciones (en promedio 5.78%) que la heurística SPS (en promedio 6.70%).

Tabla 5.4: Comparación de los métodos en el conjunto de instancias B.

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
100	5	G1	94	299.51	13.54	7.00	0.33	0.00	0.00	2.13	1.80	1.06
100	5	G2	94	100.38	16.60	7.09	0.33	0.00	0.00	2.13	1.83	0.00
100	10	G3	83	60.76	123.02	14.02	0.67	0.00	0.00	8.43	8.68	4.82
100	10	G4	84	81.22	42.57	15.21	0.74	0.00	0.00	7.14	8.47	5.95
150	10	G5	95	484.01	295.30	40.55	0.90	0.00	0.00	7.37	4.45	2.11
150	10	G6	96	190.53	215.18	36.85	0.88	0.00	0.00	7.29	4.23	2.08
150	15	G7	89	238.66	98.38	49.78	1.16	0.00	0.00	8.99	8.00	5.62
150	15	G8	89	300.10	416.93	47.88	1.07	0.00	0.00	10.11	8.77	5.62
Promedio Total				219.40	152.69	27.30	0.76	0.00	0.00	6.70	5.78	3.41

En el conjunto de instancias C (Tabla 5.5), se observa que el mejor método exacto toma en promedio cerca de 4 minutos, mientras que la heurística propuesta toma menos de 3 segundos. Cuando comparamos QR con SPS, se puede observar que QR es más rápida y encuentra soluciones de mejor calidad significativamente.

Tabla 5.5: Comparación de los métodos en el conjunto de instancias C.

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
100	10	SJC1	364	235.20	333.99	14.59	0.54	0.00	0.00	26.67	25.52	5.28
200	15	SJC2	304	34.55	82.46	79.90	1.09	0.00	0.00	19.02	5.88	1.60
300	25	SJC3a	278	122.07	182.95	234.15	2.31	0.00	0.00	36.94	8.97	3.24
300	30	SJC3b	253	89.86	153.86	247.68	2.66	0.00	0.00	30.86	8.23	3.27
402	30	SJC4a	284	891.59	991.84	631.14	3.37	0.00	0.00	42.29	8.58	4.51
402	40	SJC4b	239	166.30	170.73	505.12	4.36	0.00	0.00	34.93	9.15	3.60
Promedio Total				256.60	319.31	285.43	2.39	0.00	0.00	31.79	11.06	3.58

Los resultados del conjunto de instancias D-* son presentados en las Tablas 5.6–5.9. Como fue discutido en previas secciones, este conjunto es considerado complejo de resolver debido al valor de p que tiene un efecto significativo en el rendimiento de los métodos. Cuando analizamos la clase D- α (Tabla 5.6) se observa que QR supera a SPS en términos de calidad de la solución y en esfuerzo computacional. Respecto a la clase D- β (Tabla 5.7), SPS muestra un número significativo de soluciones no factibles, mientras que QR ofrece soluciones factibles para todas las instancias. Acerca de las clases D- γ y D- δ (Tablas 5.8–5.9) el nivel de infactibilidad de SPS se incrementa considerablemente, mientras que QR aporta soluciones factibles para todas las instancias la clase D- γ y sólo obtiene dos soluciones infactibles en el conjunto D- δ .

Tabla 5.6: Comparación de los métodos en el conjunto de instancias D- α .

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
50	5	01	29	0.09	1.61	0.70	0.22	0.00	0.00	0.00	0.00	0.00
		02	33	0.88	3.53	0.98	0.20	0.00	0.00	0.00	0.00	0.00
		03	26	0.15	1.50	0.90	0.22	0.00	0.00	7.69	0.00	0.00
		04	32	0.82	2.56	1.01	0.21	0.00	0.00	0.00	0.00	0.00
		05	29	2.01	3.34	1.11	0.23	0.00	0.00	0.00	1.04	0.00
		06	31	2.72	3.68	1.05	0.22	0.00	0.00	3.23	3.18	0.00
		07	30	1.04	2.45	1.22	0.23	0.00	0.00	0.00	0.93	0.00
		08	31	0.85	2.85	1.08	0.24	0.00	0.00	0.00	2.49	0.00
		09	28	2.20	3.11	1.30	0.23	0.00	0.00	0.00	3.50	0.00
		10	32	3.27	4.69	2.51	0.22	0.00	0.00	9.38	12.75	0.00
Promedio				1.40	2.93	1.19	0.22	0.00	0.00	2.03	2.39	0.00
100	10	11	19	11.11	3.83	9.18	0.56	0.00	0.00	5.26	11.37	0.00
		12	20	1.27	4.41	10.21	0.54	0.00	0.00	15.00	5.30	0.00
		13	20	8.73	4.13	9.67	0.54	0.00	0.00	10.00	1.75	0.00
		14	20	2.77	2.99	8.68	0.54	0.00	0.00	10.00	3.40	0.00
		15	21	6.96	5.45	9.60	0.55	0.00	0.00	9.52	4.19	0.00
		16	20	22.41	10.97	10.34	0.55	0.00	0.00	5.00	4.80	0.00
		17	22	97.78	552.17	9.78	0.57	0.00	0.00	4.55	5.86	4.55
		18	21	12.26	8.58	8.21	0.56	0.00	0.00	9.52	2.38	0.00
		19	21	6.58	13.05	10.20	0.56	0.00	0.00	9.52	6.85	0.00
		20	21	423.63	18.72	8.87	0.59	0.00	0.00	9.52	9.52	9.52
Promedio				59.35	62.43	9.48	0.56	0.00	0.00	8.79	5.54	1.41
150	15	21	16	23.46	5.48	27.48	0.98	0.00	0.00	25.00	11.25	6.25
		22	17	116.45	1351.99	24.76	1.01	0.00	0.00	11.76	4.83	0.00
		23	16	14.79	18.03	23.35	1.03	0.00	0.00	18.75	11.88	0.00
		24	16	82.06	28.03	29.18	1.01	0.00	0.00	25.00	12.63	6.25
		25	16	1.17	8.75	22.99	0.95	0.00	0.00	12.50	0.00	0.00
		26	16	11.34	28.41	28.48	0.97	0.00	0.00	12.50	8.88	6.25
		27	18	1363.65	82.05	27.22	1.08	0.00	0.00	11.11	10.28	5.56
		28	17	15.84	9.78	28.06	0.96	0.00	0.00	5.88	0.00	0.00
		29	15	12.65	4.22	25.69	0.94	0.00	0.00	20.00	13.06	6.67
		30	15	9.73	7.41	28.87	0.96	0.00	0.00	20.00	6.80	6.67
Promedio				165.12	154.41	26.61	0.99	0.00	0.00	16.25	7.96	3.77
200	20	31	14	21.69	7.15	52.55	1.50	0.00	0.00	21.43	5.57	0.00
		32	14	3600.00	1670.61	86.42	1.80	23.08	0.00	28.57	27.94	14.29
		33	14	30.85	17.82	57.16	1.43	0.00	0.00	28.57	11.07	7.14
		34	15	1134.60	88.45	60.03	1.70	0.00	0.00	26.67	12.81	6.67
		35	14	84.64	52.33	66.88	1.58	0.00	0.00	14.29	10.91	7.14
		36	14	30.71	23.50	57.54	1.53	0.00	0.00	14.29	10.00	7.14
		37	14	26.43	5.14	65.99	1.55	0.00	0.00	28.57	7.00	0.00
		38	14	743.73	245.75	66.66	1.63	0.00	0.00	28.57	18.18	7.14
		39	13	74.08	140.99	56.29	1.48	0.00	0.00	23.08	12.77	7.69
		40	15	512.57	123.18	66.56	1.57	0.00	0.00	20.00	6.54	0.00
Promedio				625.93	237.49	63.61	1.58	2.31	0.00	23.40	12.28	5.72
Promedio Total				212.95	114.32	25.22	0.84	0.58	0.00	12.62	7.04	2.72

Por último, los resultados de conjuntos de datos E se muestran en la Tabla 5.10. Aquí se observa que los métodos exactos no logran alcanzar optimalidad para ninguna instancia durante las 6 horas de ejecución. En este sentido, el ADF proporciona la mejor cota inferior (f^{lb}) para todos los casos, en contraste con OP que no reporta ninguna cota inferior válida. En cuanto a la calidad de la solución heurística, el método propuesto obtiene mejores soluciones que la heurística de SPS en un

Tabla 5.7: Comparación de los métodos en el conjunto de instancias D- β .

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
50	12	01	19	1.53	4.85	1.18	0.43	0.00	0.00	0.00	5.16	0.00
		02	20	2.06	5.49	2.19	0.42	0.00	0.00	10.00	13.70	10.00
		03	20	2.90	5.20	1.96	0.47	0.00	0.00	10.00	11.40	5.00
		04	24	3.22	6.25	2.51	0.49	0.00	0.00	8.33	20.75	16.67
		05	21	54.74	7.66	3.28	0.45	0.00	0.00	9.52	21.48	14.29
		06	22	11.60	5.71	3.73	0.41	0.00	0.00	4.55	16.00	0.00
		07	20	4.38	3.00	4.99	0.41	0.00	0.00	0.00	10.39	0.00
		08	23	4.94	5.28	3.78	0.39	0.00	0.00	4.35	4.35	4.35
		09	19	9.55	4.51	0.45	0.35	0.00	0.00	-	22.74	21.05
		10	23	1.70	7.35	0.15	0.32	0.00	0.00	-	21.43	4.35
Promedio				9.66	5.53	2.42	0.42	0.00	0.00	5.84	14.74	7.57
100	25	11	14	244.17	433.85	8.72	1.48	0.00	0.00	0.00	14.57	0.00
		12	13	4.18	5.81	9.67	1.40	0.00	0.00	15.38	11.28	7.69
		13	13	2045.22	115.48	9.13	1.51	0.00	0.00	23.08	19.96	0.00
		14	14	3600.00	25.21	10.12	1.50	25.00	0.00	7.14	7.26	7.14
		15	14	7.46	10.16	10.07	1.54	0.00	0.00	21.43	20.80	14.29
		16	14	44.50	10.92	11.56	1.44	0.00	0.00	7.14	12.72	7.14
		17	14	115.64	11.90	12.55	1.45	0.00	0.00	21.43	26.98	21.43
		18	14	15.74	7.83	10.68	1.42	0.00	0.00	14.29	22.29	0.00
		19	13	79.23	12.34	14.01	1.30	0.00	0.00	30.77	38.37	30.77
		20	*13	3600.00	3600.02	0.58	1.10	90.00	7.69	-	81.80	38.46
Promedio				975.61	423.35	9.71	1.41	11.50	0.77	15.63	25.60	12.69
150	37	21	11	428.62	13.45	36.69	2.92	0.00	0.00	27.27	27.07	18.18
		22	11	220.74	21.29	26.91	2.68	0.00	0.00	18.18	34.95	18.18
		23	*11	3600.00	3600.00	28.26	2.64	30.00	10.00	27.27	45.86	27.27
		24	11	729.86	119.67	26.09	2.97	0.00	0.00	18.18	17.63	9.09
		25	10	5.57	8.62	15.78	2.64	0.00	0.00	20.00	16.40	10.00
		26	10	8.82	13.89	33.45	2.51	0.00	0.00	30.00	21.40	10.00
		27	12	166.86	41.04	213.91	2.07	0.00	0.00	25.00	68.84	33.33
		28	11	125.18	9.61	31.11	2.51	0.00	0.00	18.18	28.13	9.09
		29	10	5.15	9.31	28.81	2.70	0.00	0.00	20.00	17.60	10.00
		30	10	33.94	23.38	28.23	2.60	0.00	0.00	30.00	23.60	10.00
Promedio				532.48	386.03	46.92	2.62	3.00	1.00	23.41	30.15	15.51
200	50	31	9	1050.13	104.27	48.09	6.55	0.00	0.00	22.22	23.11	22.22
		32	*9	3600.00	3600.04	9.27	4.64	62.50	11.11	-	287.13	144.44
		33	*9	3600.00	3600.02	71.79	7.16	25.00	12.50	44.44	44.44	44.44
		34	*10	3600.00	1857.22	25.50	4.76	22.22	11.11	-	105.89	60.00
		35	10	3600.00	39.94	42.46	6.86	37.50	0.00	20.00	20.28	10.00
		36	9	1572.58	192.25	60.29	7.02	0.00	0.00	33.33	33.76	22.22
		37	9	69.72	26.71	45.71	6.67	0.00	0.00	22.22	19.81	11.11
		38	10	3600.00	348.39	63.14	7.28	37.50	0.00	20.00	25.39	10.00
		39	9	405.74	21.53	46.96	6.39	0.00	0.00	22.22	14.44	11.11
		40	9	3600.00	1662.99	55.01	7.50	37.50	0.00	33.33	53.56	33.33
Promedio				2469.82	1145.34	46.82	6.48	22.22	3.47	27.22	62.78	36.89
Promedio Total				996.89	490.06	26.47	2.73	9.18	1.31	18.03	33.32	18.17

tiempo más corto. En particular, QR es considerablemente más rápida que cualquier otro método. La disminución en la calidad de la solución de los métodos heurísticos es notable, pero el esfuerzo de cálculo es considerablemente menor que cualquier otro método existente. La Tabla 5.9 resume la comparación entre todos los métodos. Analizando esta tabla, se observa en general que QR es considerablemente más rápida que SPS para todos los conjuntos de instancias. Respecto a la calidad de la

Tabla 5.8: Comparación de los métodos en el conjunto de instancias D- γ .

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
50	16	01	17	12.23	7.56	1.23	0.55	0.00	0.00	0.00	8.70	0.00
		02	18	19.24	4.37	2.67	0.61	0.00	0.00	5.56	12.13	5.56
		03	15	2.12	9.27	1.99	0.55	0.00	0.00	0.00	47.33	0.00
		04	21	3.62	6.00	3.40	0.64	0.00	0.00	4.76	14.29	14.29
		05	18	0.36	5.75	3.07	0.63	0.00	0.00	0.00	32.96	16.67
		06	19	39.46	4.20	6.69	0.53	0.00	0.00	5.26	32.00	15.79
		07	18	13.92	4.59	7.20	0.57	0.00	0.00	5.56	22.37	5.56
		08	20	13.08	9.13	0.53	0.52	0.00	0.00	-	28.17	20.00
		09	17	5.32	4.32	0.62	0.48	0.00	0.00	-	33.43	11.76
		10	23	718.80	157.68	0.19	0.49	0.00	0.00	-	38.00	13.04
		Promedio		82.82	21.29	2.76	0.56	0.00	0.00	3.02	26.94	10.27
100	33	11	11	9.84	7.49	12.61	2.47	0.00	0.00	27.27	30.65	9.09
		12	12	22.21	11.94	12.73	2.20	0.00	0.00	8.33	28.50	8.33
		13	12	1067.97	9.16	9.68	2.03	0.00	0.00	8.33	19.67	0.00
		14	12	1251.07	15.25	16.82	2.10	0.00	0.00	16.67	28.17	16.67
		15	13	49.37	8.84	11.29	2.14	0.00	0.00	7.69	22.82	7.69
		16	13	3600.00	31.83	17.15	2.20	18.18	0.00	15.38	38.12	15.38
		17	13	481.70	16.79	24.23	1.95	0.00	0.00	7.69	12.69	0.00
		18	14	3600.00	13.84	18.27	2.14	111.11	0.00	7.14	22.38	7.14
		19	*11	3600.00	3600.00	28.15	2.39	30.00	10.00	18.18	67.09	18.18
		20	12	121.80	60.46	4.76	1.67	0.00	0.00	-	95.79	41.67
		Promedio		1380.39	377.56	15.57	2.13	15.93	1.00	12.96	36.59	12.42
150	50	21	11	3600.00	1486.52	58.20	5.79	22.22	0.00	18.18	49.09	18.18
		22	10	1547.93	35.63	59.59	5.00	0.00	0.00	30.00	120.00	70.00
		23	11	3600.00	51.95	32.27	4.33	22.22	0.00	-	114.90	63.64
		24	10	3600.00	45.94	44.21	5.84	37.50	0.00	20.00	37.20	30.00
		25	9	118.75	10.82	24.29	6.07	0.00	0.00	11.11	32.66	11.11
		26	9	24.30	11.59	26.44	5.93	0.00	0.00	22.22	54.14	22.22
		27	*11	1986.46	3600.00	22.10	4.22	100.00	10.00	-	159.90	54.55
		28	10	3600.00	132.94	28.33	5.70	37.50	0.00	10.00	32.61	10.00
		29	9	5.34	123.35	30.29	5.69	0.00	0.00	22.22	47.22	33.33
		30	9	263.13	110.06	20.30	6.18	0.00	0.00	22.22	48.00	22.22
		Promedio		1834.59	560.88	34.60	5.47	21.94	1.00	19.49	69.57	33.53
200	66	31	8	423.45	119.78	52.80	10.04	0.00	0.00	25.00	40.49	12.50
		32	9	3600.00	1437.69	9.73	6.94	42.86	0.00	-	393.83	244.44
		33	8	3600.00	1411.95	67.81	10.29	42.86	0.00	37.50	64.65	50.00
		34	10	2931.58	813.40	67.84	7.17	166.67	0.00	-	185.17	110.00
		35	8	253.08	309.73	65.16	10.52	0.00	0.00	37.50	54.38	37.50
		36	8	3600.00	390.48	40.23	11.42	42.86	0.00	37.50	62.29	37.50
		37	8	10.38	104.06	58.08	10.50	0.00	0.00	25.00	56.32	25.00
		38	*8	3600.00	3600.02	73.45	10.07	42.86	14.29	50.00	70.42	50.00
		39	8	16.22	114.41	46.90	10.06	0.00	0.00	25.00	27.92	25.00
		40	9	3600.00	160.53	70.91	11.02	42.86	0.00	22.22	54.26	33.33
		Promedio		2163.47	846.20	55.29	9.80	38.10	1.43	32.47	100.97	62.53
		Promedio Total		1365.32	451.48	27.05	4.49	18.99	0.86	16.99	58.52	29.68

solución, QR provee soluciones aceptables para los conjuntos de instancias A, B, C y los subconjuntos D- α y D- β . Para los subconjuntos D- γ y D- δ , la comparación entre QR y SPS no tiene un gran sentido por que SPS falla en encontrar soluciones factibles en un número considerable de instancias. Analizando el conjunto de instancias E, el rendimiento de la QR es mejor que todos los demás métodos, cuyo tiempo y recursos de cálculo crece considerablemente, alcanzando factibilidad en todos los

Tabla 5.9: Comparación de los métodos en el conjunto de instancias D- δ .

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
50	20	01	17	3.49	115.99	1.78	0.64	0.00	0.00	0.00	62.23	5.88
		02	18	2.69	67.85	4.24	0.95	0.00	0.00	5.56	59.93	16.67
		03	17	3.35	81.86	6.29	0.80	0.00	0.00	0.00	130.85	41.18
		04	20	3600.00	143.61	4.71	0.74	11.11	0.00	10.00	44.00	20.00
		05	19	21.13	78.39	0.36	0.59	0.00	0.00	-	88.76	42.11
		06	21	148.47	26.64	0.22	0.60	0.00	0.00	-	60.93	40.00
		07	19	34.36	24.75	0.23	0.65	0.00	0.00	-	70.10	10.53
		08	23	3600.00	154.27	0.23	0.59	26.32	0.00	-	39.48	4.35
		09	21	3600.00	99.31	0.22	0.58	10.53	0.00	-	92.28	42.86
		10	23	17.94	72.06	0.24	0.60	0.00	0.00	-	159.78	82.61
Promedio				1103.14	86.47	1.85	0.67	4.80	0.00	3.89	80.83	30.62
100	40	11	12	62.32	119.26	37.62	2.77	0.00	0.00	16.67	76.48	41.67
		12	11	3600.00	103.47	25.44	3.18	20.00	0.00	18.18	74.30	36.36
		13	12	339.46	130.37	22.88	2.71	0.00	0.00	0.00	73.80	41.67
		14	12	1479.71	90.34	70.71	2.21	0.00	0.00	-	121.62	58.33
		15	13	2.64	165.71	5.78	2.35	0.00	0.00	-	130.92	69.23
		16	13	328.78	179.14	103.63	2.23	0.00	0.00	-	66.41	30.77
		17	14	3600.00	172.38	5.49	2.10	16.67	0.00	-	134.98	64.29
		18	14	3600.00	175.17	3.39	2.02	16.67	0.00	-	111.15	50.00
		19	12	3600.00	284.53	29.40	2.15	183.33	0.00	-	109.83	58.33
		20	14	3600.00	489.86	3.93	2.07	16.67	0.00	-	278.57	164.29
Promedio				2021.29	191.02	30.83	2.38	25.33	0.00	11.62	117.81	61.49
150	60	21	11	3600.00	223.20	242.81	8.12	220.00	0.00	18.18	119.80	63.64
		22	12	3600.00	2942.79	10.23	5.20	20.00	0.00	-	275.37	116.67
		23	11	1151.41	198.78	5.87	5.20	0.00	0.00	-	273.67	127.27
		24	9	3464.17	148.95	81.27	6.88	0.00	0.00	33.33	120.43	66.67
		25	9	4.69	109.68	24.77	7.85	0.00	0.00	11.11	73.89	22.22
		26	9	49.30	127.01	36.84	10.51	0.00	0.00	22.22	86.36	44.44
		27	*13	3600.00	3600.00	7.74	5.08	27.27	11.11	-	238.46	238.46
		28	10	3600.00	1546.36	35.98	7.47	25.00	0.00	20.00	72.17	30.00
		29	9	4.94	145.64	38.47	8.56	0.00	0.00	22.22	91.05	44.44
		30	9	8.84	105.95	31.32	7.58	0.00	0.00	11.11	77.22	33.33
Promedio				1908.33	914.84	51.53	7.24	29.23	1.11	19.74	142.84	78.71
200	80	31	8	35.47	198.93	55.32	13.82	0.00	0.00	37.50	75.00	37.50
		32	*10	3391.43	1522.98	11.98	8.45	433.33	37.50	-	-	-
		33	8	34.10	206.54	107.37	12.38	0.00	0.00	37.50	112.57	62.50
		34	12	3600.00	268.32	11.98	8.73	325.00	0.00	-	-	-
		35	*8	3600.00	1528.03	125.24	11.40	400.00	14.29	50.00	124.79	62.50
		36	9	58.11	157.17	403.80	11.61	0.00	0.00	-	135.37	66.67
		37	8	11.74	156.87	72.87	14.14	0.00	0.00	25.00	109.10	62.50
		38	8	3600.00	266.42	18.90	9.34	28.57	0.00	-	245.88	100.00
		39	8	75.61	136.24	42.66	17.40	0.00	0.00	25.00	80.90	37.50
		40	*8	3600.00	1161.64	94.24	11.64	400.00	14.29	-	186.80	87.50
Promedio				1800.64	560.31	94.44	11.89	158.69	6.61	35.00	133.80	64.58
Promedio Total				1708.35	438.16	44.66	5.55	54.51	1.93	17.56	118.82	58.85

casos. Respecto a factibilidad, QR sigue siendo mejor, ya que fue capaz de encontrar una solución factible en todos los casos a diferencia de SP que reporta 34 soluciones no factibles. En comparación con los métodos exactos QR es aún más confiable en términos del número de soluciones factibles encontradas. Los métodos exactos ADF y OP reportan 46 y 17 soluciones no factibles respectivamente, mientras QR reporta sólo 2 soluciones no factibles.

Tabla 5.10: Comparación de los métodos en el conjunto de instancias E

n	p	Instancia	Óptimo	Tiempo (s)				Desviación (%)				
				ADF	OP	SPS	QR	ADF	OP	SPS	QR ¹	QR ²
3038	600	R1	*77	21600.00	70.77	1038.46	192.35	93.51	-	532.47	115.54	88.31
3038	700	R2	*58	21600.00	66.86	941.10	246.38	156.90	-	406.90	177.59	165.52
3038	800	R3	*58	21600.00	67.71	2975.48	326.45	156.90	-	372.41	201.03	163.79
3038	900	R4	*58	21600.00	77.00	857.26	462.34	156.90	-	432.76	240.75	182.76
3038	1000	R5	*40	21600.00	67.83	3575.76	524.59	272.50	-	485.00	471.25	352.50
Promedio Total				21600.00	70.03	1877.61	350.42	167.34	-	445.91	241.23	190.58

Tabla 5.11: Resumen de la comparación entre los métodos para todos los conjuntos de instancias.

Conjunto	Promedio de DR (%)				Tiempo promedio (s)				Soluciones infactibles			
	ADF	OP	SPS	QR ²	ADF	OP	SPS	QR	ADF	OP	SPS	QR
A	0.00	0.00	5.41	0.23	30.07	32.69	5.34	0.39	0	0	0	0
B	0.00	0.00	6.70	3.41	219.40	152.69	27.30	0.76	0	0	0	0
C	0.00	0.00	31.79	3.58	256.60	319.31	285.43	2.39	0	0	0	0
D- α	0.58	0.00	12.62	2.72	212.95	114.32	25.22	0.84	1	0	0	0
D- β	9.18	1.31	18.03	18.17	996.89	490.06	26.47	2.73	9	5	5	0
D- γ	18.99	0.86	16.99	29.68	1365.32	451.48	27.05	4.49	14	3	8	0
D- δ	54.51	1.93	17.56	58.85	1708.35	438.16	44.66	5.55	17	4	21	2
E	167.34	-	445.91	190.58	21600.00	70.03	1877.61	350.42	5	5	0	0

5.3.3 ANÁLISIS DE COMPONENTES

En este experimento, se evalúa la aportación que cada componente brinda a la heurística QR. Se consideran los tres principales componentes del método: Perturbación, VND y Agitación. El experimento consiste en deshabilitar un componente a la vez y ejecutar la heurística QR usando 1000 iteraciones como límite para los conjuntos de instancias A, B, C y D-*, y 200 iteraciones para E. Se usan para cada conjunto de instancias los valores de α presentados en la Tabla 5.2 utilizados previamente. La Tabla 5.12 muestra la comparación de los componentes para cada conjunto de instancias.

En esta tabla la columna «Todo» representa el promedio de DR cuando todos los componentes están habilitados, el cual coincide con el valor desplegado en la Tabla 5.11, columna «QR²». En cada columna de la sección «Componentes» representa el componente deshabilitado durante el experimento y muestra el valor obtenido.

Tabla 5.12: Análisis de los componentes dentro de la heurística QR.

Conjunto	Todo	Componentes			Contribución (%) a $f(X)$		
		Perturbación	VND	Agitar	Perturbación	VND	Agitar
A	0.23	17.25	2.77	2.70	77.25	11.52	11.23
B	3.41	12.53	4.66	5.33	74.19	10.16	15.65
C	3.58	36.63	8.32	15.59	66.38	9.51	24.12
D- α	2.72	31.87	5.61	5.39	83.98	8.32	7.70
D- β	18.17	106.97	33.19	29.04	77.42	13.09	9.48
D- γ	29.68	195.23	46.81	45.03	83.60	8.65	7.75
D- δ	58.85	261.94	105.76	92.62	71.57	16.53	11.90
E	190.58	496.62	245.65	192.91	84.20	15.15	0.64

La sección de «Contribución %» muestra el valor porcentual que cada componente específico aporta al valor total desplegado en «Todo». Es notable como el componente que tiene mayor influencia en la heurística es Perturbación, seguido por VND y Agitar. Esto demuestra ser consistente con el análisis estadístico, que mostraba que el parámetro α , usado en la perturbación, influye en la variable de respuesta. Sin embargo, los otros dos componentes añaden valor al rendimiento total del método. El beneficio de VND oscila desde 9.59 % a 31.26 %, y el beneficio aportado por el mecanismo de Agitar oscila desde 0.64 % a 27.12 %. En conclusión, todos los componentes del método son importantes.

5.3.4 RECURSOS COMPUTACIONALES

Basado en los resultados del experimento anterior procedemos a realizar una comparación asintótica de los recursos computacionales. La Figura 5.3 muestra una comparación asintótica entre los métodos en términos del tiempo de ejecución y uso de memoria con respecto al número de nodo. De manera similar la Figura 5.4 muestra una comparación asintótica pero con respecto al número de nodo y valor de p , donde el tamaño del círculo expresa el tiempo de cómputo en segundos, y el color el uso de memoria en bytes. Las estadísticas de memoria indican el máximo tamaño del conjunto residente utilizado en bytes [55], esto es, el máximo número de bytes de la memoria física que cada proceso usa simultáneamente.

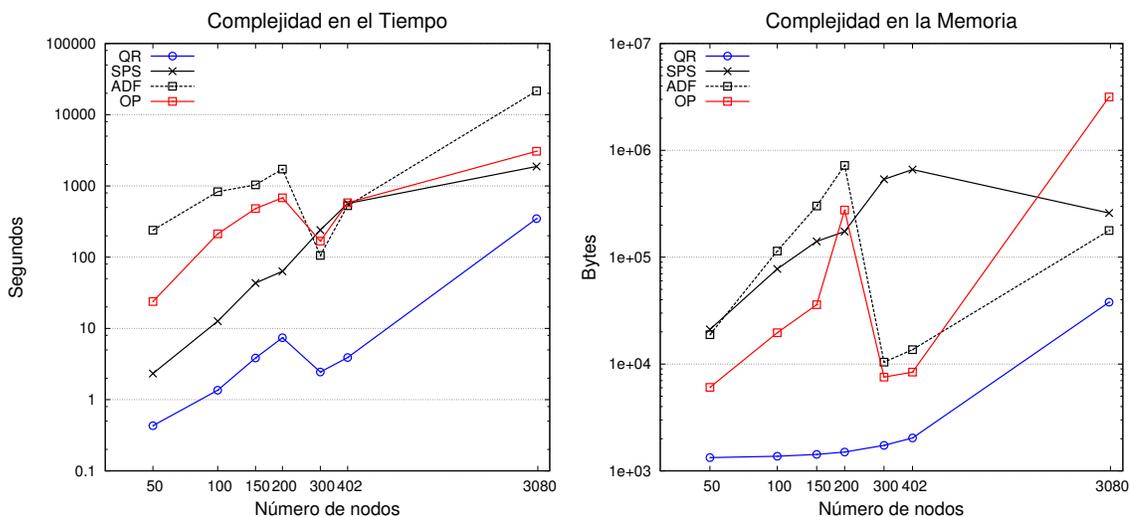


Figura 5.3: Comparación asintótica en función de n entre los métodos en términos de tiempo de ejecución y uso de memoria. Los ejes se encuentran en escala logarítmica.

En la Figura 5.3 se observa que el método propuesto requiere menos recursos para su ejecución a diferencia de los otros tres métodos conforme el número de nodos incrementa. El crecimiento sigue el comportamiento esperado de la cota asintótica determinada en la Sección 3.2. En particular el uso de memoria requerida por los dos métodos exactos, como es de esperarse, es considerablemente mayor.

Como es sabido, el valor de p influye significativamente en el rendimiento de los métodos. La Figura 5.4 muestra el análisis asintótico de recursos en función de n y p . Aquí se observa como el método propuesto logra mantener un uso de recursos moderado conforme n y p incrementan, principalmente en memoria, no así la heurística SPS cuyos niveles de memoria se elevan considerablemente cuando el valor de p es elevado en proporción al tamaño de la instancia, característica de los problemas de las instancias D y E. Respecto a los métodos exactos, en función del tiempo, OP pareciera ser el más robusto en términos de tiempo y memoria para las instancias del conjunto A, B, C y D, sin embargo, el uso de memoria crece considerablemente para las instancias del conjunto E. Por el contrario, ADF, mantiene un uso de memoria razonable, lo cual le permitió aportar las cotas inferiores para este conjunto.

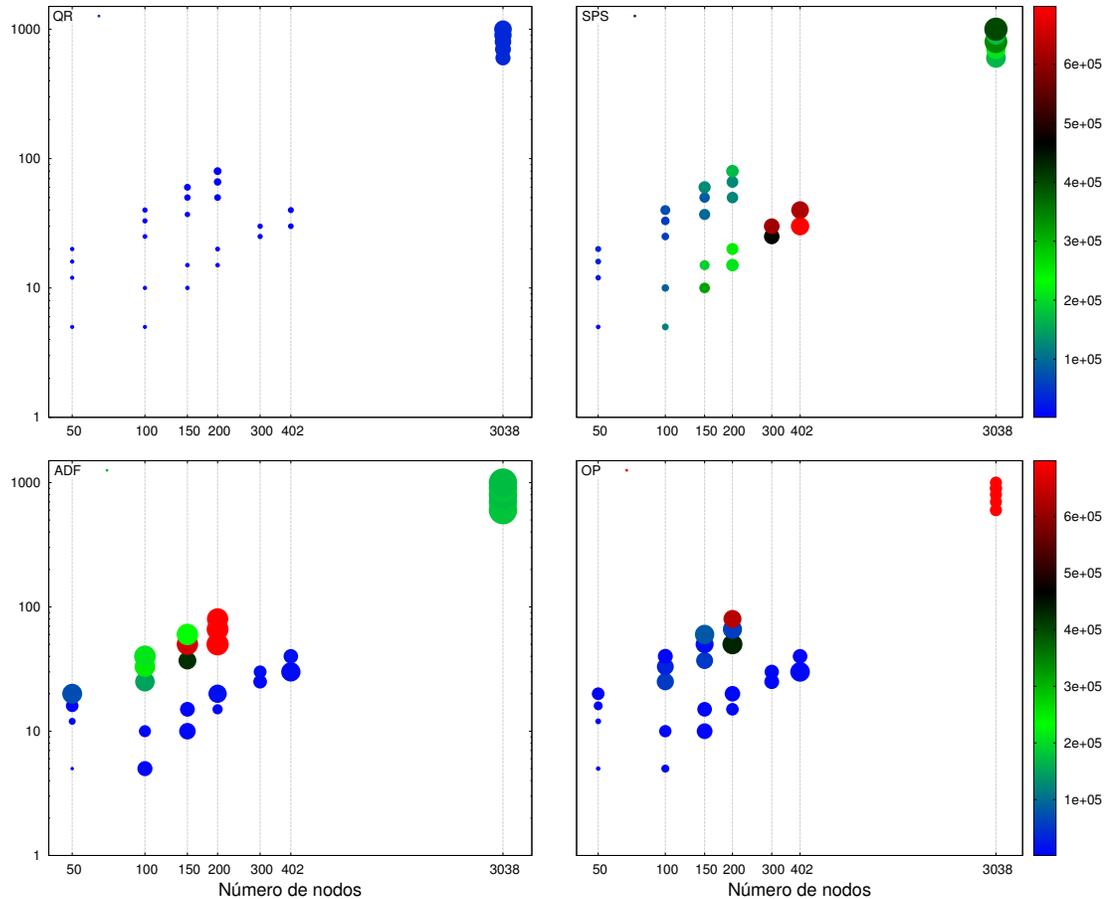


Figura 5.4: Comparación asintótica en función de n y p entre los métodos en términos de tiempo de ejecución y uso de memoria. El eje vertical expresa el valor de p , el tamaño del círculo indica el tiempo de cómputo en segundos (s) con una proporción de $\log(s + 2)$ y el color el uso de memoria en bytes. Los ejes se encuentran en escala logarítmica.

5.4 EVALUACIÓN DEL MÉTODO EXACTO

En esta sección se reporta la evaluación del método exacto BP&C. Se ejecuta una experimentación computacional del método basada únicamente en los conjuntos A, B y C; finalmente se muestran, comparan y argumentan los resultados obtenidos, los cuales son determinantes para aportar conclusiones razonables del método. Tal como fue definido, el método requiere una cota inferior y superior que en este caso son calculadas a través de la primera fase del método ADF [4], el cual resulta ser más

eficiente entre los métodos exactos para el cálculo de cotas iniciales. Las condiciones computacionales son definidas en la Sección 5.2.

Las Tablas 5.13–5.15 presentan los resultados de la experimentación. En cada tabla, las dos primeras columnas representan el tamaño de la instancia medido por el número de nodos n y el número de particiones p . «Instancia» es el nombre de la instancia del problema y «Óptimo» indica el valor óptimo de la instancia, valor obtenido por los métodos ADF y OP. Las columnas f^{lb} y f^{ub} son las cotas inferior y superior, respectivamente encontradas por el método. La columna «Desviación (%)» muestra la desviación porcentual relativa entre cotas alcanzado por BP&C, «Tiempo (s)¹» despliega el tiempo total en segundo utilizado, contemplando en este valor el tiempo requerido para el cálculo de las cotas y el cálculo por BP&C. De manera similar en la columna «Memoria (B)¹» expresa las estadísticas de memoria que indican el máximo tamaño del conjunto residente utilizado en bytes.

Tabla 5.13: Resultado de BP&C en la solución del conjunto de instancias A.

n	p	Instancia	Óptimo	f^{lb}	f^{ub}	Desviación	Tiempo(s) ¹	Memoria ¹
50	5	cpmp01	29	29	29	0.00	48.40	1.43E+6
		cpmp02	33	33	33	0.00	231.12	1.99E+6
		cpmp03	26	26	26	0.00	239.39	2.00E+6
		cpmp04	31	31	33	6.45	228.54	1.98E+6
		cpmp05	28	28	38	35.71	940.52	2.02E+6
		cpmp06	30	30	32	6.67	756.61	2.02E+6
		cpmp07	29	29	32	10.34	341.67	1.99E+6
		cpmp08	31	31	31	0.00	204.94	2.00E+6
		cpmp09	28	28	28	0.00	428.46	1.99E+6
		cpmp10	32	32	32	0.00	375.65	2.00E+6
		Promedio				5.92	379.53	1.94E+6
100	10	cpmp11	19	19	22	15.79	126.18	2.22E+6
		cpmp12	19	19	20	5.26	126.14	2.17E+6
		cpmp13	19	19	21	10.53	124.96	2.14E+6
		cpmp14	20	20	21	5.00	124.87	2.14E+6
		cpmp15	20	20	24	20.00	125.39	2.19E+6
		cpmp16	20	20	20	0.00	124.22	2.19E+6
		cpmp17	21	21	24	14.29	454.56	2.15E+6
		cpmp18	21	21	24	14.29	121.41	2.15E+6
		cpmp19	20	20	21	5.00	124.80	2.18E+6
		cpmp20	19	19	23	21.05	122.21	2.17E+6
		Promedio				11.12	157.47	2.17E+6
		Promedio Total				8.52	268.50	2.06E+6

Respecto al conjunto de instancias A (Tabla 5.13), el método es en promedio eficiente en calidad para el grupo de instancias pequeñas (tamaño 50×5), alcanzando óptimalidad en 6 instancias, pero obteniendo un valor de tiempo y memoria bastante elevado. Para el grupo de mayor tamaño (tamaño 100×10) sólo se alcanza óptimalidad en una instancia; este conjunto obtiene una desviación promedio de 8.52 %. Para el conjunto de instancias B (Tabla 5.14) se observa que a pesar de que no se encuentra el óptimo en todas las instancias, se tiene una desviación promedio de 6.38 %, todas menor al 12 %, sin embargo el incremento en los recursos computacionales es considerable en recursos computacionales

Tabla 5.14: Resultado de BP&C en la solución del conjunto de instancias B.

n	p	Instancia	Óptimo	f^{lb}	f^{lub}	Desviación	Tiempo(s) ¹	Memoria ¹
100	5	G1	93	93	99	6.06	424.17	2.24E+6
100	5	G2	92	92	96	4.17	249.80	2.23E+6
100	10	G3	77	77	86	10.47	160.39	2.24E+6
100	10	G4	79	79	86	8.14	137.77	2.30E+6
150	10	G5	94	94	97	3.09	234.54	2.69E+6
150	10	G6	94	94	97	3.09	207.57	2.63E+6
150	15	G7	86	86	93	7.53	181.70	2.62E+6
150	15	G8	86	86	94	8.51	171.30	2.62E+6
Promedio Total						6.38	220.90	2.45E+6

Tabla 5.15: Resultado de BP&C en la solución del conjunto de instancias C.

n	p	Instancia	Óptimo	f^{lb}	f^{lub}	Desviación	Tiempo(s) ¹	Memoria ¹
100	10	SJC1	320	320	423	24.35	177.34	2.13E+6
200	15	SJC2	301	301	323	6.81	34.23	3.17E+6
300	25	SJC3a	273	273	320	14.69	56.29	3.82E+6
300	30	SJC3b	248	248	270	8.15	62.02	3.84E+6
402	30	SJC4a	276	276	332	16.87	104.80	5.27E+6
402	40	SJC4b	237	237	423	43.97	108.57	5.26E+6
Promedio Total						19.14	90.54	3.91E+6

Finalmente, en el conjunto de instancias C (Tabla 5.15) se tiene una desviación promedio de 15.28 %, sin embargo no se encuentra el óptimo en ninguna instancia y en tres de las instancias no se reportaron cotas válidas. El consumo de recursos, sobre todo la memoria, se incrementó considerable. Cabe señalar que el criterio de

paro satisfecho para todas las instancias con un valor desviación mayor a cero, fue rebasar el límite de memoria establecido en 4 GB.

5.4.1 RECURSOS COMPUTACIONALES

Basado en los resultados del experimento anterior se realizar una comparación asintótica de los recursos computacionales. La Figura 5.5 muestra el tiempo de ejecución asintótica y uso de memoria con respecto al número de nodos. No se realiza una comparación respecto a los demás métodos propuestos pues no tendría un fundamento equitativo, debido a que BP&C no reportó soluciones óptimas en casi la totalidad de las instancias de prueba, sin embargo la gráfica nos ayuda a interpretar el rápido crecimiento en recursos. Conforme la memoria incrementa, decrementa el tiempo, esto debido al criterio de paro por memoria que delimita sustancialmente el tiempo de ejecución del método.

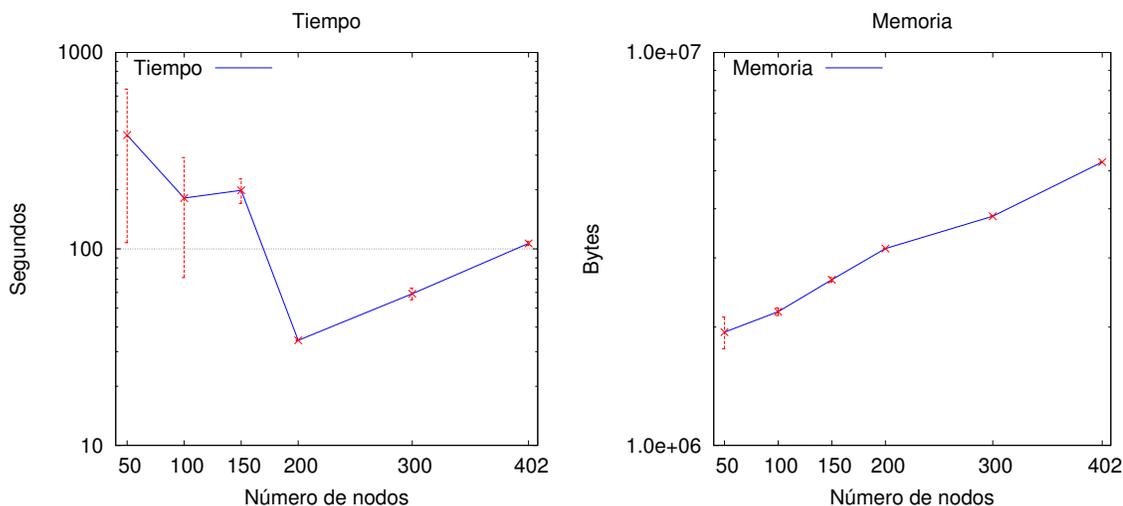


Figura 5.5: Gráfica asintótica en términos de tiempo de ejecución y uso de memoria de BP&C. Los ejes verticales se encuentran en escala logarítmica. La barra vertical expresa la desviación estándar.

El método parece tener un rendimiento eficiente y alcanzar óptimalidad con instancias de tamaño $n \leq 50$. Experimentos no incluidos en este trabajo con instancias con tamaño entre 5 y 40 nodos muestra que el método alcanza óptimalidad

sin mayor problema. Sin embargo, no es competitivo frente a otros métodos exactos previamente propuestos en la literatura como Özsoy y Pınar [69] (OP) y Albareda-Sambola, Díaz y Fernández [4] (ADF).

Una conclusión preliminar de la posible falla de la estrategia radica en la función objetivo minimax, que es convexa lineal por partes. Este tipo de problemas es en sí mismo una serie de subproblemas de minimización.

El método depende, para garantizar una óptima convergencia, de cotas iniciales con una desviación muy estrecha para acotar el dominio de la variable de linealización z . Si las cotas no cumplen con estas características el rendimiento del subproblema de precio se ve decrementado, dado que en cada iteración se genera un número elevado de columnas factibles dentro del rango actual de la variable z , por lo que el tiempo requerido para la resolución del problema maestro restringido, relajación lineal y obtención de valores duales se incrementa considerablemente. En consecuencia se ocasiona una lenta convergencia del método de generación de columnas, que en el peor caso termina con un desbordamiento de la memoria, tal como sucede en el anterior experimento. Otra consecuencia de esto, es la expansión del árbol de solución. Los criterios seleccionados para realizar la ramificación, limitados por la función objetivo, impiden implementar una estrategia que limite sustancialmente el árbol de solución tales como reducir la simetría en el espacio de soluciones del problema.

CONCLUSIONES Y CONTRIBUCIONES

6.1 MÉTODO HEURÍSTICO

En el presente trabajo se ha propuesto un procedimiento de solución heurística compuesto por una estrategia de búsqueda voraz aleatorizada con selección probabilística en la fase de construcción y un IGLS con VND en su fase de mejora o búsqueda local. A partir del estudio computacional presentado en el Capítulo 5, se concluye que la heurística propuesta supera a la única heurística conocida para este problema en la literatura (SPS) tanto en términos de calidad de la solución, tiempo de solución y factibilidad. El procedimiento propuesto resulta ser más robusto que los métodos exactos al obtener soluciones con valores objetivos de calidad para los conjuntos de instancias A, B, C, D- α y D- β , además de requerir menos recursos computacionales. Para el conjunto de instancias clasificado como difícil de resolver, en particular D- γ , D- δ y E, el valor objetivo de la solución heurística no resulta ser favorable; sin embargo, el método obtiene una solución de manera rápida. Cabe señalar que el procedimiento propuesto proporcionó soluciones factibles para la gran mayoría de las instancias probadas, obteniendo sólo un 1 % de instancias infactibles, lo que es claramente superior a la heurística SPS, ya que esta última falla en el 17 % de instancias.

Respecto al análisis de los componentes, se ha evidenciado que el logro de la heurística se debe principalmente al métodos de Perturbación; sin embargo los métodos de VND y Agitar resultaron de gran utilidad en los conjuntos A, B y C, al

aportar mejora al valor de la función objetivo. En conclusión todos los componentes implementados añaden valor positivo al rendimiento total del método.

Es claro notar la sencillez del método, al no definir estructuras y rutinas complejas dentro de sus componentes, siendo esto la esencia de los algoritmos IGLS, lo cual permite lograr una eficiente implementación computacional en un lenguaje de alto nivel. El método propuesto requiere sólo dos parámetros para su ejecución, siendo α el que impacta más al rendimiento del heurístico y por lo tanto necesita un análisis más minucioso para fijar un valor. Sin embargo, tal como se realizó en el presente trabajo, un diseño experimental basado en un único factor (α) resulta una tarea sencilla y proporciona una certeza estadística sobre la elección de un valor determinado para α .

En conclusión, el método propuesto provee soluciones robustas en un menor tiempo y recursos computacionales para los problemas discutidos previamente en la literatura, para el conjunto D y E presentando en este trabajo, el método garantiza una mayor factibilidad y rapidez comparado con la heurística existente, lo cual representa un avance al estado del arte en este campo del conocimiento. Por todo lo anterior y en base a la evidencia empírica presentada se acepta la Hipótesis 1.

6.2 MÉTODO EXACTO

En el presente trabajo se han sentado las bases de un procedimiento de solución exacto a través de una estrategia de BP&C. A partir del estudio computacional presentado en el Capítulo 5, se observa que el método exacto reporta soluciones factibles en 31 de las 34 instancias de prueba, de las cuales sólo en el 22 % se alcanza la óptimalidad. Si bien el porcentaje no es muy amplio, para el conjunto de instancias A se alcanza una desviación promedio de 8.52 %. Con respecto al conjunto B se logra una desviación promedio de 6.38 %. Con respecto al conjunto C, se logra una desviación promedio 15.28 %. En este último conjunto no se logran reportar soluciones factible en tres de las instancias y se observa un consumo de recursos

computacionales bastante elevado.

En comparación con los otros métodos de solución propuestos, BP&C no aporta soluciones factibles que aseguren óptimalidad. Los métodos exactos propuestos por Özsoy y Pınar [69] y Albareda-Sambola, Díaz y Fernández [4] proveen una mayor garantía de alcanzar soluciones óptimas. Además, el procedimiento propuesto asume la existencia de cotas de buena calidad para delimitar el número de variables y el dominio de la variable de linealización z , por lo cual su rendimiento es dependiente de la calidad de otros procedimientos para la obtención de cotas.

El principal motivo de la limitación del método es atribuido a la estructura del problema. La función objetivo minimax (la cual es convexa lineal por partes) del problema restringe en buena medida el uso de ciertos paradigmas de métodos exactos, BP&C quizá no sea la excepción. Hasta donde se conoce en la literatura no se ha reportado algún método de generación de columnas para problemas con función objetivo minimax, al menos de forma directa sobre un modelo original. Rodrigues-Lima [81] propone un método de generación de columnas para un problema con función objetivo minimax, sin embargo, él separa el problema en k subproblemas, que se resuelven de forma individual por generación de columnas; sin embargo depende de un valor de cota inicial de buena calidad que asegure la rápida convergencia del método.

Tal como se ha mencionado, el método depende de cotas iniciales con una desviación muy estrecha para acotar el dominio de la variable de linealización z y así garantizar una óptima convergencia. Si las cotas no cumplen con esta característica el rendimiento del subproblema de precio se ve decrementado, dado que en cada iteración se genera un número elevado de columnas factibles dentro del rango actual de la variable z . Por tal motivo el tiempo requerido para la resolución del problema maestro restringido se incrementa considerablemente lo que produce una lenta convergencia del método de generación de columnas. Por todo lo anterior y en base a la evidencia empírica presentada se rechaza la Hipótesis 2.

6.3 TRABAJO A FUTURO

Respecto al método heurístico, algunos componentes y funciones pueden adoptar el paradigma de programación paralela. Por ejemplo, el cálculo de los centros de una p -partición es procedimiento que se puede paralelizar dado que la operación es independiente en cada partición. Otra operaciones potenciales a adoptar este esquema es durante el cálculo de probabilidades de las funciones (3.3) y (3.6) para cada nodo $j \in V$. Para los tamaños de instancias de hasta 3038 nodos y 1000 instalaciones presentados en este trabajo, el paradigma de operaciones secuenciales utilizado fue suficiente y competitivo para resolver con recursos computacionales moderados; sin embargo se propone resolver bajo este enfoque instancias con tamaño mucho mayores y no reportados aún en la literatura.

Los componentes que integran el método propuesto se han diseñado de forma que puedan ser adaptado y extendidos sin mucho esfuerzo según se requiera para problemas con función objetivo similar al $CpCP$ e incluso otros problemas de localización tales como el $CpMP$. Como se ha demostrado en el presente trabajo, una estrategia heurística que combina IGLS con VND resulta favorables para este tipo de problemas de localización. En el caso de $CpMP$, el cambio inicial es adaptar las medidas de desempeño que evalúa el nivel de factibilidad en una asignación, definidas en la función (3.4), la cual contemplaría que la asignación disminuyera la suma de las distancias de cada partición, de igual manera la fase de perturbación incluiría un criterio de desconexión de aquellos elementos cuya distancia incrementa la suma de la partición. Se plantea como trabajo a futuro adoptar la estrategia utilizada aplicando a su vez el paradigma de programación paralela a un problema de $CpMP$, realizar una evaluación empírica del método y comparar con los métodos heurísticos encontrados en la literatura para este problema.

APÉNDICES

APÉNDICE A

MARCO TEÓRICO

En este apéndice se describen las bases teóricas, terminología y notación que se usan en esta tesis (definiciones que pueden encontrarse en los textos de Ahuja, Magnanti y Orlin [3], Lawler [53] y Papadimitriou y Steiglitz [71]). Se abordan los conceptos básicos de teoría de grafos, aspectos de la optimización combinatoria y la complejidad computacional.

A.1 TEORÍA DE GRAFOS

Definición A.1 Un *grafo* G es un par $G = (V, E)$, donde V es un conjunto finito de *nodos* o *vértices* y E tiene subconjuntos de cardinalidad dos llamados *aristas* [71]. Los nodos en V se denotan por v_1, v_2, \dots, v_n , donde $n = |V|$. Por ejemplo, el grafo

$$G = (\{v_1, v_2, v_3, v_4\}, \{[v_1, v_2], [v_2, v_3], [v_3, v_4], [v_4, v_1], [v_1, v_3]\}),$$

se muestra en la Figura A.1(a). En ocasiones es usual considerar un *multigrafo*, que es un grafo con aristas repetidas, mostrado en la Figura A.1(b). Un grafo es *planar* si puede ser dibujado en el plano sin que ninguna arista se cruce. Los grafos mostrados en la Figura A.1 son planares.

Definición A.2 Un *grafo dirigido* o *digrafo*, es un grafo con dirección asignada a cada una de las aristas. Formalmente, un digrafo D es un par $D = (V, A)$ donde V es un conjunto de nodos y A es un conjunto de *pares ordenados* de nodos llamados *arcos*; esto es, $A \subseteq V \times V$. En la Figura A.1(c), se muestra el digrafo

$$D = (\{v_1, v_2, v_3, v_4\}, \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1), (v_1, v_4), (v_1, v_3)\}).$$

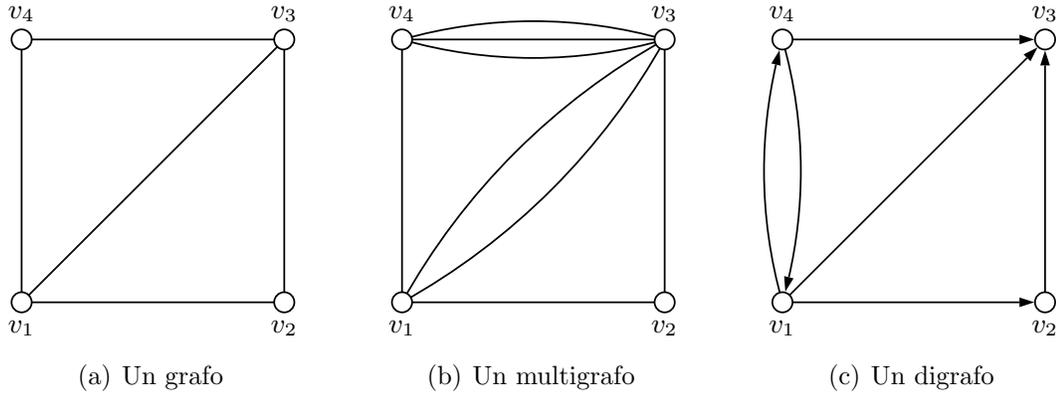


Figura A.1: Diferentes tipos de grafos.

Definición A.3 Si $G = (V, E)$ es un grafo y $e = [v_1, v_2] \in E$, entonces se dice que v_1 es *adyacente* a v_2 (y viceversa) y e es *incidente* en v_1 y v_2 . El *grado* del nodo v en G , es el número de aristas que son incidentes al nodo v . En el grafo de la Figura A.1(a), el grado de v_1 es tres. En un digrafo $D = (V, A)$ el *grado de entrada* de un nodo v es el número de arcos en la forma (u, v) que están en A . Similarmente, el *grado de salida* de v es el número de arcos en A que tienen la forma (v, u) .

Definición A.4 Sea $G = (V, E)$ un grafo, se dice que $G' = (V', E')$ es *subgrafo* de G si $E' \subseteq E$ y $V' \subseteq V$.

Definición A.5 Un *camino* en G es una secuencia de nodos $w = (v_1, v_2, v_3, \dots, v_k)$ tal que $k \geq 1$ y $(v_j, v_{j+1}) \in E$ para $j = \{1, \dots, k-1\}$. El camino es *cerrado* si $k > 1$ y $v_k = v_1$. Un camino sin nodos repetidos es llamado *ruta*. Un camino cerrado sin nodos repetidos, a excepción del primero es llamado *circuito* o *ciclo*. Por ejemplo en la Figura A.1(a) $(v_1), (v_1, v_2, v_3, v_1, v_4, v_1), (v_1, v_2, v_3, v_1)$ y (v_2, v_3, v_4) son caminos; el segundo y tercero son cerrados, y el primero y cuarto son rutas, y el tercero es un ciclo. La *longitud* de una *ruta* es $k-1$ y para un *ciclo* $(v_1, \dots, v_k = v_1)$ es k .

Definición A.6 Un grafo $G = (V, E)$ es conexo si, para cualquier par de nodos u y v en V , existe al menos una *ruta* de u a v .

A.1.1 MÉTRICAS DE DISTANCIA

Definición A.7 Un *grafo ponderado*, es un grafo $G = (V, E)$ con una función w desde E a \mathbb{Z} (usualmente \mathbb{Z}_+ ; aunque también puede ser \mathbb{R}_+ , por ejemplo cuando los pesos son distancias euclidianas). En ciertos casos es más usado un nombre mnemotécnico para los pesos, tal como c (para costos) o d (para distancias). El peso de la arista (u, v) es denotado por $w(u, v)$ o w_{uv} .

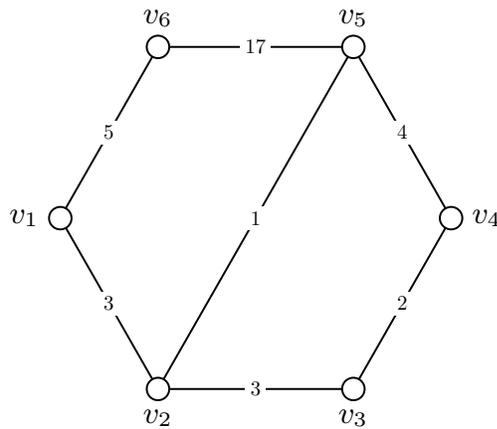


Figura A.2: Grafo planar ponderado con pesos en \mathbb{Z}_+ .

La métrica de distancia define la lejanía entre un par de nodos (u, v) . En teoría de grafos y localización es usualmente utilizada la *distancia euclidiana* o *euclídea* [26, 78], que es la longitud de la recta que une a dos puntos, en el espacio de dimensión \mathbb{R}^2 . La función de distancia entre los puntos $\mathbf{x} = (x_1, x_2, \dots, x_n)$ e $\mathbf{y} = (y_1, y_2, \dots, y_n)$ es llamada $d(\mathbf{x}, \mathbf{y})$ (que corresponde a la coordenada geográfica de cada nodo en V) y definida de la manera siguiente

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}, \quad (\text{A.1})$$

por lo tanto la distancia entre dos nodos (u, v) es

$$d_{uv} = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}; \quad (\text{A.2})$$

se dice que la distancia es simétrica si $d_{uv} = d_{vu}$, para todo $u, v \in V$. El presente trabajo asume que la distancia entre cualquier par de nodos de una instancia de prueba (ver Sección 5.2.1), es euclidiana. En caso de no existir en la instancia la distancia, pero conociendo el conjunto de coordenadas geográficas (\mathbf{x}, \mathbf{y}) , se procede a calcular de manera euclídea la distancia simétrica entre todos los pares de nodos.

A.1.2 FUNCIONES DE DISPERSIÓN

Una función de dispersión mide qué tan dispersos se encuentran entre sí los nodos de un grafo. A continuación se describen las funciones más comunes, seguido de un sencillo ejemplo de cada uno en un grafo planar presentados en la Figura A.3.

p-centro: mide la máxima de las distancias entre el centro y el resto de los nodos. Entonces, sea $c \in V$ el centro del grafo $G(V, E)$, donde la dispersión de V está dada por $\max_{j \in V} \{d_{cj}\}$. Ésta es la función de dispersión utilizada en este trabajo, la cual se denota como el problema de optimización de encontrar el nodo centro $c \in V$ que minimize esta dispersión, es decir $c = \arg \min_{i \in V} \{\max_{j \in V} \{d_{ij}\}\}$.

p-mediana: de forma similar al *p*-centro, mide la suma de las distancias entre el centro y el resto de los nodos. De esta manera, sea $c \in V$ el centro del grafo $G(V, E)$, donde la dispersión de V esta dada por $\sum_{j \in V} d_{cj}$. El problema de optimización de encontrar el nodo centro $c \in V$ que minimize esta dispersión es $c = \arg \min_{i \in V} \sum_{j \in V} d_{ij}$.

Diamétero: mide la distancia máxima entre todos los pares de nodos, por lo cual no requiere un centro en el grafo $G(V, E)$. Entonces la dispersión de V esta dada por $\max_{i \in V} \max_{j \in V} \{d_{ij}\}$.

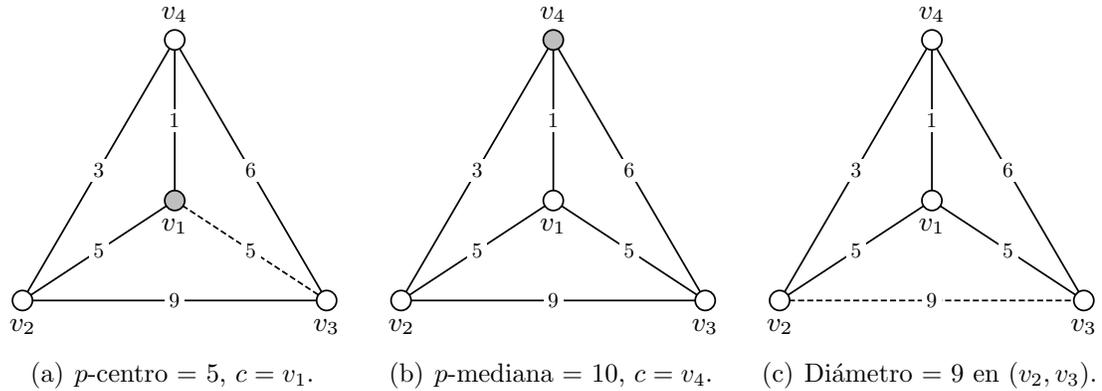


Figura A.3: Comparación de las diferentes funciones de dispersión en un grafo planar.

A.2 OPTIMIZACIÓN COMBINATORIA

En el lenguaje coloquial, optimizar significa poco más que mejorar; sin embargo, en el contexto científico la optimización es el proceso de tratar de encontrar la mejor solución posible para un determinado problema [60]. En un *problema de optimización* existen diferentes soluciones y un criterio para discriminar entre ellas. El objetivo es encontrar la mejor solución de todas, es decir, aquella cuyo valor de una función objetivo alcanza su valor máximo o mínimo. Formalmente [91], un MILP se define de la manera siguiente

$$\max_{x \in \mathbb{R}^n} \{ \mathbf{c}^\top \mathbf{x} : \mathbf{A} \mathbf{x} \leq \mathbf{b}, x_i \in \mathbb{Z}, \forall i \in I \}, \quad (\text{A.3})$$

donde $A \in \mathbb{Q}^{m \times n}$ es la matriz de restricciones, $\mathbf{c} \in \mathbb{Q}^n$ es un vector de costos, $\mathbf{b} \in \mathbb{Q}^m$ es un vector del lado derecho de las restricciones, e $I \subseteq \{1, \dots, n\}$ es el conjunto de índices de las variables que son restringidas a valores enteros. Se sabe de varios problemas MILP cuya resolución es un problema NP-duro [34]. Sin embargo, debido a los recientes avances, tanto en la teoría subyacente e implementaciones computacionales disponibles, la optimización discreta es ahora una herramienta viable para optimizar algunos de los sistemas más complejos mediante técnicas modernas de programación entera-mixta y software especializado.

Un problema MILP usualmente puede ser representado como un problema *combinatorio*, el cual se define como un conjunto específico de instancias que son objetos combinatoriales. Aquí típicamente se tiene un conjunto finito $I = \{1, \dots, n\}$, costo c_i para cada $i \in I$ y un conjunto \mathcal{S} de subconjuntos factibles de I . El problema de seleccionar el subconjunto factible que minimize el costo es

$$\text{mín} \left\{ \sum_{i \in s} c_i : s \in \mathcal{S} \right\}; \quad (\text{A.4})$$

la tarea relacionada a la instancia del problema es encontrar una solución globalmente óptima $s^* \in \mathcal{S}$ tal que $f(s^*) \leq f(s)$ para todo $s \in \mathcal{S}$. Se denota por $f^* = f(s^*)$ el costo de la solución óptima y por \mathcal{S}^* el conjunto de soluciones globalmente óptimas. Los problemas combinatoriales son usualmente fáciles de definir, pero muy difíciles de resolver. Una forma de solucionar este tipo problema, aunque no la más inteligente, es por enumeración exhaustiva. Es decir, se hace una lista completa de todas las soluciones factibles y se evalúa la función objetivo para cada una, eligiendo al final la solución que proporciona el mejor valor de ésta función. Sin embargo el número

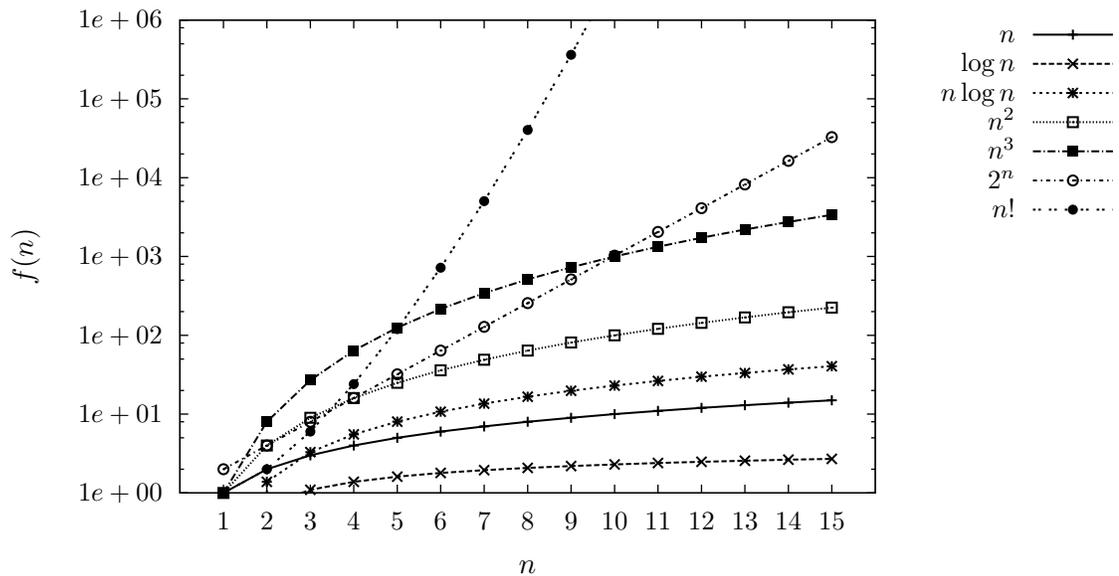


Figura A.4: Crecimiento de algunas funciones logarítmicas y exponenciales en función de un entero n , que representa el tamaño de una instancia. El eje vertical se encuentran en escala logarítmica.

de soluciones factibles suele ser enorme, por lo cual el algoritmo *ingenuo* tiene complejidad computacional exponencial, haciendo impráctica la evaluación de todas sus soluciones para determinar el óptimo. La Figura A.4 muestra el crecimiento de algunas funciones logarítmicas y exponenciales en función de un entero n , por lo cual la complejidad de un algoritmo de enumeración exhaustiva tendría al menos una complejidad de $\Omega(f(n))$. Nemhauser y Wolsey [67], y Wolsey [91] proporcionan un análisis a profundidad acerca de la teoría de la programación entera y combinatoria.

Definición A.8 Se denota por la función $\mathcal{O}(f(n))$ una *cota superior asintótica* al tiempo de ejecución de un algoritmo, mientras que $\Omega(f(n))$ es una *cota inferior asintótica*, dicho de otra manera $\mathcal{O}(f(n))$ representa el peor escenario en el tiempo de ejecución de un algoritmo y $\Omega(f(n))$ el mejor escenario.

COMPLEJIDAD COMPUTACIONAL

En términos informales, se dice que un algoritmo es *eficiente* si su tiempo de ejecución es *polinomial*. Un problema que cuenta con por lo menos un algoritmo eficiente es un problema polinomial. Un problema es *intratable* si no existe algún algoritmo eficiente para resolverlo y finalmente se dice que es un problema *sin solución* si no cuenta con ninguno algoritmo que lo resuelva.

Si no existe un algoritmo que resuelva el problema de forma polinomial se dice que el problema es *difícil* de resolver o NP-duro. Esto no implica que el problema no pueda resolverse, sino que cada algoritmo existente para la solución del problema tiene un tiempo de ejecución que crece exponencialmente con el tamaño del problema. Un *problema de decisión* es aquél donde las únicas respuestas posibles son sí o no; para demostrar que un problema es difícil, este es transformado en un problema de decisión y se prueba que es NP-completo y por lo tanto el problema de optimización es NP-duro. Cook [14], Garey y Johnson [34] y Papadimitriou y Steiglitz [71, 70] realizan un detallado análisis de la teoría acerca de la complejidad computacional.

A.2.1 MÉTODOS DE SOLUCIÓN HEURÍSTICOS

Dada la dificultad práctica para resolver de forma exacta toda una serie de importantes problemas combinatorios para los cuales, por otra parte es importante ofrecer alguna solución, comenzaron a aparecer algoritmos que proporcionan soluciones factibles, es decir, que satisfacen todas las restricciones del problema. Este tipo de algoritmos se denominan *heurísticas* [19], del griego *heuriskein*, «encontrar». Aunque en un primer momento no fueron bien vistas en los círculos académicos, acusadas de escaso rigor matemático [22], su interés práctico como herramienta útil que da soluciones a problemas reales, les fue abriendo poco a poco las puertas, sobre todo a partir de la mitad de los años setenta con la proliferación de resultados en el campo de la complejidad computacional. La Figura A.5 muestra un cronograma y genealogía de las heurísticas con mayor impacto en el estado del arte. Las heurísticas son procedimientos simples, a menudo basados en el sentido común, que se supone ofrecen una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido. Muchas de las estrategias de búsqueda podrían usarse, aunque algunas pueden resultar ser demasiado costosas o quedar atrapadas en óptimos locales.

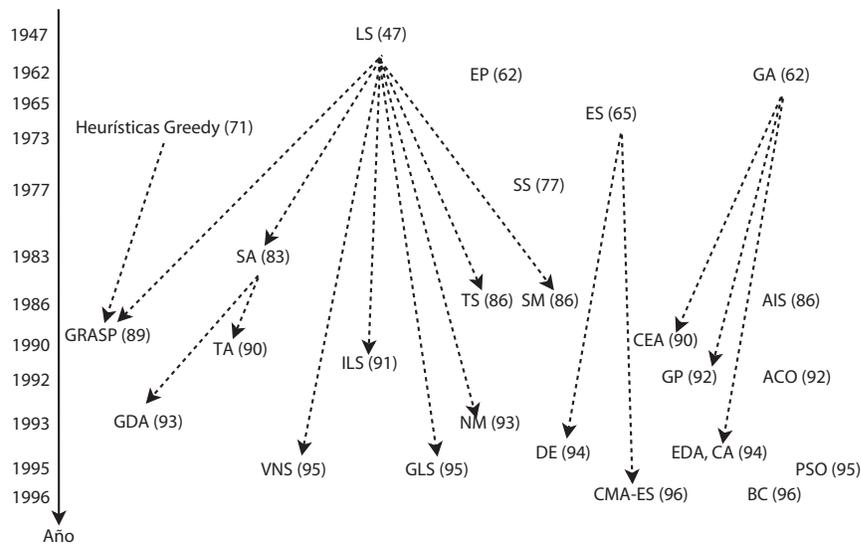


Figura A.5: Cronograma y genealogía de las heurísticas.

Los procedimientos *metaheurísticos* [49] son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria. Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos. Se proponen como métodos, deterministas o estocásticos para salir de óptimos locales. Se ha mencionado el caso de la resolución de un problema difícil como la aplicación práctica de métodos heurísticos, sin embargo existen otras razones para utilizar métodos heurísticos.

- El problema es de una naturaleza tal que no se conoce ningún método exacto para su resolución.
- Aunque existe un método exacto para resolver el problema, su uso es computacionalmente muy costoso.
- El método heurístico es más flexible que un método exacto, permitiendo, por ejemplo, la incorporación de condiciones de difícil modelización.

Al abordar el estudio de los algoritmos heurísticos y metaheurísticos podemos comprobar que dependen en gran medida del problema concreto para el que se han diseñado. En otros métodos de resolución de propósito general, como puede ser un algoritmo exacto, se realiza un procedimiento conciso y preestablecido independiente en gran medida del problema abordado. En los métodos heurísticos esto no es así, las técnicas e ideas aplicadas a la resolución de un problema son específicas de éste y aunque, en general, pueden ser trasladadas a otros problemas, han de particularizarse en cada caso [60].

APÉNDICE B

MÉTODOS DE DESCOMPOSICIÓN

En este apéndice se describen algunos de los métodos de descomposición tradicionales y compuestos, además de la terminología y notación relacionadas en otros apartados de esta tesis. Esta sección fue basada en el texto de Wolsey [91] y en los tutoriales de Ralphs y Galati [74], Galati [29], y Galati, Ralphs y Wang [31].

Los métodos de descomposición son técnicas que explotan las subestructuras tratables de un MILP con el fin de obtener mejores técnicas de solución. En particular, la idea fundamental es explotar la capacidad para optimizar ya sea separando la envolvente convexa de soluciones, a través de una relajación con el fin de obtener mejores métodos que acoten el valor de la solución óptima. Estos métodos de acotamiento puede ser entonces combinados con métodos de ramificación de tal manera que se obtengan robustos algoritmos basados en B&B.

La manera más común para realizar esta explotación de las subestructuras es relajar un conjunto de «restricciones difíciles». Este es el enfoque adoptado por Dantzig-Wolfe, Lagrange y método de planos de corte. Las estructuras también se pueden exponer mediante la fijación de los valores de un conjunto de variables, teniendo en cuenta las restricciones del problema original; este es el enfoque adoptado por la descomposición de Benders. A continuación se revisan los principios de este tipo de técnicas.

B.1 PRINCIPIOS BÁSICOS

Para facilitar la explicación se asume un programa lineal entero puro (ILP, por sus siglas en inglés: *integer linear program*) con una cota inferior y superior finita para todas las variables, por lo tanto el conjunto de soluciones factibles es finito. Para el resto de la explicación consideraremos una instancia ILP cuyo conjunto factible es un vector de enteros contenidos en el poliedro

$$\mathcal{Q} = \{x \in \mathbb{R}^n : Ax \geq b\}, \quad (\text{B.1})$$

donde $A \in \mathbb{Q}^{m \times n}$ es una matriz de restricciones y $b \in \mathbb{R}^m$ es vector del lado derecho. Sea $\mathcal{F} = \mathcal{Q} \cap \mathbb{Z}^n$ el conjunto de soluciones factibles del ILP y \mathcal{P} la envolvente convexa de \mathcal{F} . En términos de esta notación, el ILP es determinado como un programa entero (IP, por sus siglas en inglés: *integer program*)

$$z_{IP} = \min_{x \in \mathcal{F}} \{c^\top x\} = \min_{x \in \mathcal{P}} \{c^\top x\} = \min_{x \in \mathbb{Z}^n} \{c^\top x : Ax \geq b\}, \quad (\text{B.2})$$

donde $c \in \mathbb{R}^n$. Por convención, $z_{IP} = \infty$ si $\mathcal{F} = \emptyset$, es decir el problema IP es infactible. Se denota por $OPT(\mathcal{P}, c, U)$ una subrutina que resuelve el problema de decisión, donde U es una cota superior. Esta rutina devolverá un conjunto de elementos de \mathcal{P} con valor objetivo mejor que U . Para aplicar los principios que rigen la descomposición, se considera la relajación de IP definida por

$$\min_{x \in \mathcal{F}'} \{c^\top x\} = \min_{x \in \mathcal{P}'} \{c^\top x\} = \min_{x \in \mathbb{Z}^n} \{c^\top x : A'x \geq b'\}, \quad (\text{B.3})$$

donde $\mathcal{F}' \subset \mathcal{F} = \{x \in \mathbb{Z}^n : A'x \geq b'\}$ para algún $A' \in \mathbb{Q}^{m' \times n}$, $b' \in \mathbb{Q}^{m'}$ y \mathcal{P}' es la envolvente convexa de \mathcal{F}' . Aquí se asume que existe al menos un algoritmo eficiente que realiza la separación de \mathcal{P}' . Se denota por $SEP(\mathcal{P}, x)$ una subrutina que separa un vector $x \in \mathbb{R}^n$ del poliedro \mathcal{P} retornando un conjunto de desigualdades válidas violadas. La forma de optimización de un problema de separación es el de encontrar la desigualdad que genere una mayor violación. Con respecto a \mathcal{F}' , sea

$[A'', b''] \in \mathbb{Q}^{m'' \times n''}$ el conjunto de restricciones adicionales, necesarias para describir a \mathcal{F} , es decir, $[A'', b'']$ es tal que $\mathcal{F} = \{x \in \mathbb{Z}^n : A'x \geq b', A''x \geq b''\}$. Se denota por \mathcal{Q}' el poliedro que define las desigualdades $[A', b']$ y por \mathcal{Q}'' el poliedro que define las desigualdades $[A'', b'']$. Por lo tanto, la relajación inicial LP es el programa lineal definido por $\mathcal{Q} = \mathcal{Q}' \cap \mathcal{Q}''$ y la cota LP es dada por

$$z_{LP} = \min_{x \in \mathcal{Q}} \{x^\top x\} = \min_{x \in \mathbb{R}^n} \{c^\top x : A'x \geq b', A''x \geq b''\}. \quad (\text{B.4})$$

Note que $[A', b']$ y $[A'', b'']$ son frecuentemente una partición de renglones de $[A, b]$ dentro del conjunto de «restricciones agradables» y un conjunto de «restricciones difíciles», respectivamente; sin embargo no es un requerimiento estricto. La Figura B.1(a) (obtenida originalmente de [74]) muestra un ejemplo de los diversos poliedros, el conjunto de soluciones factibles es $\mathcal{F} = \mathcal{Q}' \cap \mathcal{Q}'' \cap \mathbb{Z}^2 = \mathcal{P}' \cap \mathcal{Q}'' \cap \mathbb{Z}^2$ y $\mathcal{P} = \text{conv}(\mathcal{F})$. La Figura B.1(b) representa una aproximación continua $\mathcal{Q}' \cap \mathcal{Q}''$, mientras que la Figura B.1(c) muestra la mejora de la optimización dada por $\mathcal{P}' \cap \mathcal{Q}''$.

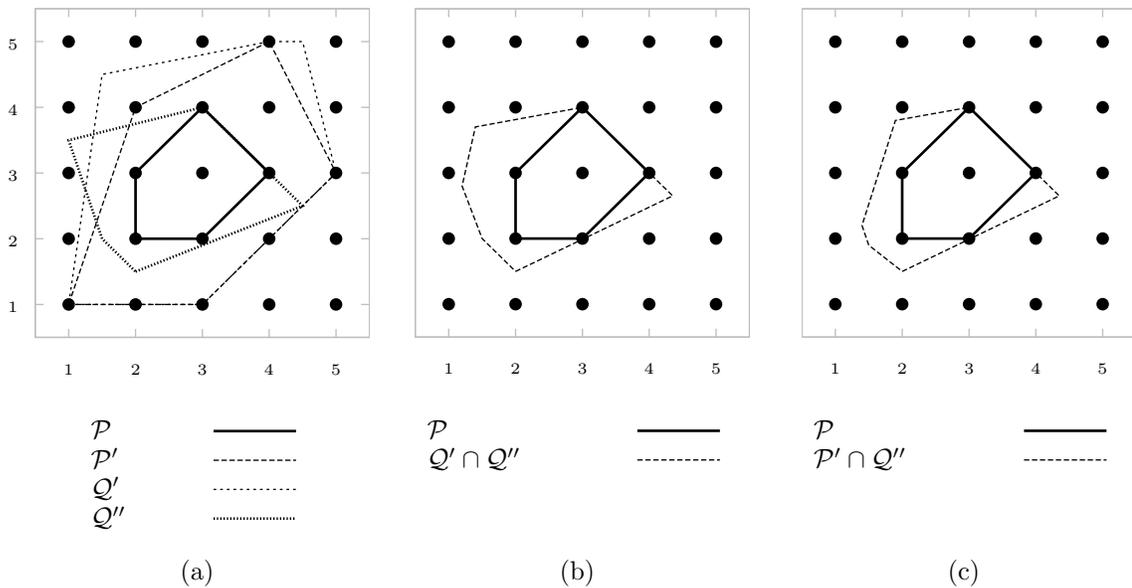


Figura B.1: Poliedros en un espacio \mathbb{R}^2 .

Realizar la optimización o separación sobre \mathcal{P}' puede ser más tratable que en \mathcal{P} por varias razones. Primero, el problema resultante tiene una estructura conocida

que puede ser analizada combinatorialmente y para el cual se tiene alguna técnica de solución bien desarrollada. El primer caso se plantea con frecuencia cuando el problema original es formado a partir de añadir restricciones laterales a un problema base bien estudiado.

$$\begin{pmatrix} A'_1 & & & & & \\ & A'_2 & & & & \\ & & \ddots & & & \\ & & & A'_{k-1} & & \\ & & & & A'_k & \\ A''_1 & A''_2 & \dots & A''_{k-1} & A''_k & \end{pmatrix}$$

Figura B.2: Estructura de los bloques diagonales de una matriz A'

En segundo lugar, el subproblema resultante puede descomponerse de forma natural, debido a la estructura de los bloques diagonales de la matriz A' . En este caso, la optimización sobre \mathcal{P}' puede ser lograda optimizando cada uno de los bloques de forma independiente (posiblemente en paralelo). Está técnica es particularmente eficaz cuando los bloques resultantes de la descomposición son idénticos, ya que esto indica cierto grado de simetría en el modelo original. La Figura B.1 muestra un ejemplo de esta estructura en una matriz A' .

B.2 MÉTODOS

Una vez identificada, la descomposición puede ser realizada utilizando tres clases de método generales. Los tres métodos calculan lo que genéricamente es llamado *cota de descomposición* computada por

$$z_D = \max_{x \in \mathcal{P}' \cap \mathcal{Q}''} c^\top x. \quad (\text{B.5})$$

Es evidente observar que $z_{LP} \leq z_D \leq z_{IP}$ y el objetivo es elegir una descomposición para la cual la desigualdad izquierda es estricta, por lo que obtenemos una mejora en la cota mediante la resolución de la relajación de LP.

B.2.1 DANTZIG-WOLFE

Este método [15] calcula la cota dinámicamente construyendo una aproximación interna a \mathcal{P}' , con la finalidad de resolver el llamado *problema maestro*, que es calculado como

$$z_{DW} = \min_{\lambda \in \mathbb{R}_+^{\mathcal{F}'}} \left\{ c^\top \left(\sum_{s \in \mathcal{F}'} s \lambda_s \right) : A'' \left(\sum_{s \in \mathcal{F}'} s \lambda_s \right) \geq b'', \sum_{s \in \mathcal{F}'} \lambda_s = 1 \right\}. \quad (\text{B.6})$$

Esto se realiza generando iteradamente elementos de \mathcal{F}' (usualmente puntos extremos de \mathcal{P}') con los cuales se forman las columnas de (B.6). En cada iteración, una columna con costo reducido negativo es generada. El problema de generación de columnas es un problema de optimización sobre \mathcal{P}' el cual denotaremos como *subproblema de generación de columnas*. El método finaliza cuando ninguna columna con costo reducido negativo puede ser encontrada. La solución de (B.6) puede ser interpretada como los pesos por el cual los miembros de \mathcal{F}' se pueden combinar para producir una solución óptima, dada por $\hat{x}_{DW} = \sum_{s \in \mathcal{F}'} s \lambda_s$ para (B.5).

B.2.2 LAGRANGE

Este método puede ser considerado como un método para resolver el dual de (B.6). Dado un vector de multiplicadores duales $u \in \mathbb{R}_+^{m''}$, la relajación Lagrangiana [27, 10, 54] de IP esta dada por

$$z_{LR}(u) = \min_{s \in \mathcal{F}'} \{ (c^\top - u^\top A'') s + u^\top b'' \} \quad (\text{B.7})$$

Se demuestra que $z_{LR}(u)$ es una cota inferior de z_{IP} . El problema

$$z_{LD} = \max_{u \in \mathbb{R}_+^{m''}} \{ z_{LR}(u) \}, \quad (\text{B.8})$$

de maximizar esta cota sobre las opciones de los multiplicadores duales es un dual para IP llamado *dual lagrangiano* (LD) y que además provee un cota inferior z_{LD} , la cual llamaremos cota LD. Un vector de multiplicadores \hat{u} que obtiene el máximo

valor de la cota es denominado *multiplicadores óptimos (duales)*. La función $z_{LR}(u)$ es linealmente cóncava a u y su máximo por lo tanto puede ser encontrado de manera eficiente por varios métodos para maximizar funciones cóncavas. Uno de los más recientes métodos es propuesto por Narciso y Lorena [65] para la solución del problema general de asignación (GAP, por sus siglas en inglés: *generalized assignment problem*).

B.2.3 PLANOS CORTANTES

Este procedimiento [91, 45] se basa en la construcción (parcial) de una aproximación exterior de \mathcal{P}' para calcular una cota de la siguiente forma

$$z_{CP} = \min_{x \in \mathcal{P}'} \{c^\top x : A''x \geq b''\}. \quad (\text{B.9})$$

Se observa que \hat{x}_{DW} , es una solución óptima para CP. Se puede resolverse de forma iterativa mediante la resolución de un *problema de separación* de \mathcal{P}' en cada iteración. Es bien sabido que el problema de la separación de un poliedro es polinomialmente equivalente al problema de optimización del mismo poliedro. Geoffrion [36] demostró la igualdad de estas tres cotas con las cotas de descomposición.

Teorema B.1 $z_{IP} \geq c^\top \hat{x}_{DW} = z_{LD} = z_{DW} = z_{CP} = \min \{c^\top x : \mathcal{P}' \cap \mathcal{Q}''\} \geq z_{LP}$.

Estos tres algoritmos se pueden abstraer en un marco algorítmico común. En los tres casos, el problema de acotamiento es un procedimiento iterativo que consta de una alternancia entre la solución de un *problema maestro restringido*, resuelto para obtener información de la solución, y un *subproblema*, resuelto para obtener información adicional acerca de la estructura poliédrica de \mathcal{P}' , la cual es utilizada para aumentar el *problema maestro*. Se continúa iterando hasta que el criterio de paro es alcanzando.

B.2.4 GENERACIÓN DE COLUMNAS CON PLANOS CORTANTES

La integración del método de planos cortantes con el método de Dantzig-Wolfe resulta en un procedimiento que alterna entre un subproblema que genera una mejora en las columnas (subproblema de precio) y un subproblema que genera un buen conjunto de desigualdades válidas (subproblema de planos cortantes). Por lo tanto el método resultante es llamado generación de columnas con planos cortes (P&C por sus siglas en inglés: *price-and-cut*). Cuando el método es implementado en un marco de B&B, la técnica es denominada ramificación, generación de columnas con planos cortes (BP&C por sus siglas en inglés: *branch-price-and-cut*). Este método ha sido estudiado por diversos autores: Savelsbergh [84], Barnhart et al. [8], Barnhart, Hane y Vance [7], Vanderbeck [89] y más recientemente Poggi de Aragão y Uchoa [16], Ralphs, Ladányi y Saltzman [76], Desrosiers y Lübbecke [18] y Gamrath [33].

Similar que en el método Dantzig-Wolfe, la cota producida por P&C puede ser pensado como el resultado de la intersección de dos poliedros que se aproximan. En Dantzig-Wolfe se requiere uno de estos \mathcal{Q}'' , para tener una breve descripción; pero estos poliedros pueden tener un tamaño exponencial. P&C permite desarrollar descripciones parciales de ambos poliedros: un poliedro interior \mathcal{P}_I y poliedro exterior \mathcal{P}_O que son generados dinámicamente. Para optimizar la intersección entre \mathcal{P}_I y \mathcal{P}_O , se usa una reformulación de Dantzig-Wolfe como en (B.6), excepto que $[A'', b'']$ es reemplazada por una matriz que cambia dinámicamente en cada iteración. El método de P&C se muestra en el Algoritmo B.2.4.

Al examinar los pasos de este método, la cuestión más interesante que surge es como los métodos de generación de columnas y desigualdades válidas son adaptados a este nuevo entorno dinámico. Sin embargo, potencialmente problemático es el hecho de que el método de generación de columnas resulta en una reducción de la cota \bar{z}_{PC}^t calculada por (B.12), por el contrario la generación de desigualdades válidas esta orientada a aumentarla. La cota \bar{z}_{PC}^t es obtenida por resolver (B.12), mientras que \underline{z}_{PC}^t es obtenida por resolver el subproblema de precio, cuyo resultado

Algoritmo 8 Método de generación de columnas y cortes**Require:** Una instancia $ILP(\mathcal{P}, c)$ 1: $t \leftarrow 0$ 2: **Inicialización:** Construir una aproximación inicial interna desde un conjunto inicial \mathcal{E}^t de puntos extremos de \mathcal{P}'

$$\mathcal{P}_I^t = \left\{ \sum_{s \in \mathcal{E}^t} s \lambda_s : \sum_{s \in \mathcal{E}^t} \lambda_s = 1, \lambda_s \geq 0 \quad \forall s \in \mathcal{E}^t, \lambda_s = 0 \quad \forall s \in \mathcal{E} \setminus \mathcal{E}^t \right\} \subseteq \mathcal{P}'. \quad (\text{B.10})$$

y una aproximación inicial externa donde $D^t = A''$ y $d^t = b''$,

$$\mathcal{P}_O^t = \{x \in \mathbb{R}^n : D^t x \geq d^t\} \supseteq \mathcal{P}. \quad (\text{B.11})$$

3: $m^t \leftarrow m'', \mathcal{E}' \leftarrow \mathcal{E}^t, [D', d'] \leftarrow [D^t, d^t]$ 4: **while** $\mathcal{E}' \neq \emptyset$ y $[D', d'] \neq \emptyset$ **do**5: **Problema maestro restringido** : Resolver la reformulación de Dantzig-Wolfe de el LP sobre el poliedro $\mathcal{P}_I^t \cap \mathcal{P}_O^t$ para obtener el valor óptimo $\bar{z}_{PC}^t = \sum_{s \in \mathcal{E}} s(\lambda_{PC}^t)_s$ y una solución óptima dual $(u_{PC}^t, \alpha_{PC}^t) \in \mathbb{R}^{m^t+1}$

$$\bar{z}_{PC}^t = \min_{\lambda \in \mathbb{R}_+^{\mathcal{E}}} \left\{ c^\top \left(\sum_{s \in \mathcal{E}} s \lambda_s \right) : D^t \left(\sum_{s \in \mathcal{E}} s \lambda_s \right) \geq d^t, \sum_{s \in \mathcal{E}} \lambda_s = 1, \lambda_s = 0 \quad \forall s \in \mathcal{E} \setminus \mathcal{E}^t \right\}. \quad (\text{B.12})$$

6: **Hacer** (a) o (b)(a) **Subproblema de precio:** Resolver $OPT(\mathcal{P}', c^\top - (u_{PC}^t)^\top D^t, \alpha_{PC}^t)$ y generar un conjunto $\mathcal{E}' \subseteq \mathcal{E}$, con elementos con costo reducido negativo, calculado para cada $s \in \mathcal{E}$ por

$$rc(s) = (c^\top - (u_{PC}^t)^\top A'') s - \alpha_{PC}^t. \quad (\text{B.13})$$

Si $\mathcal{E}' \neq \emptyset$, entonces $\mathcal{E}^{t+1} \leftarrow \mathcal{E}^t \cup \mathcal{E}'$. Si $s' \in \mathcal{E}'$ con el menor costo reducido, entonces $\underline{z}_{PC}^t = rc(s') + \alpha_{PC}^t + (d^t)^\top u_{PC}^t$ provee una cota inferior válida. Actualizar $[D^{t+1}, d^{t+1}] \leftarrow [D^t, d^t], \mathcal{P}_O^{t+1} \leftarrow \mathcal{P}_O^t, m^{t+1} \leftarrow m^t$.(b) **Subproblema de planos cortantes:** Resolver $SEP(\mathcal{P}, x_{PC}^t)$ para generar un conjunto de desigualdades válidas $[D', d'] \in \mathbb{R}^{m' \times n+1}$ para \mathcal{P} , violadas por x_{PC}^t . Si encuentra alguna desigualdad es violada, entonces $[D^{t+1}, d^{t+1}] \leftarrow [D^t, d^t] \cup [D', d']$, y formar una nueva aproximación externa \mathcal{P}_O^{t+1} . Actualizar $m^{t+1} \leftarrow m^t + m', \mathcal{E}^{t+1} \leftarrow \mathcal{E}^t, \mathcal{P}_I^{t+1} \leftarrow \mathcal{P}_I^t$.7: $t \leftarrow t + 1$ 8: **end while**9: $z_{PC} = \bar{z}_{PC} = \underline{z}_{PC}, \hat{\lambda}_{PC} = \lambda_{PC}^t, (\hat{u}_{PC}, \hat{\alpha}_{PC}) = (u_{PC}^t, \alpha_{PC}^t), [D_{PC}, d_{PC}] = [D^t, d^t]$.10: **return** Una cota inferior z_{PC} de la solución óptima de la instancia $ILP(\mathcal{P}, c)$, una solución prima óptima $\hat{x}_{PC} \in \mathbb{R}^n$, una descomposición óptima $\hat{\lambda}_{PC} \in \mathbb{R}_+^{\mathcal{E}}$, una solución dual óptima $(\hat{u}_{PC}, \hat{\alpha}_{PC}) \in \mathbb{R}^{m''+1}$ y las desigualdades $[D_{PC}, d_{PC}] \in \mathbb{R}^{m^t \times (n+1)}$.

podría resultar más relevante para el objetivo, y además esta cota puede potencialmente ser mejorada a través de generación de desigualdades válidas o columnas. Las columnas pueden ser generadas de manera similar a como se realiza en el método de Dantzig-Wolfe. Para buscar nuevas columnas, se localizan las que tienen un costo reducido negativo, su cálculo se define en la función (B.13). El método finaliza cuando no se encontraron elementos con costo reducido y no es posible generar desigualdades válidas. La Figura B.2.4 muestra el diagrama de flujo cuando el método es implementado en un marco de B&B.

B.3 SOFTWARE

El desarrollo de un software para la implementación de métodos de descomposición se ha perseguido a menudo como una meta difícil de alcanzar. Los intentos más notables son aquellos que apoyan técnicas como la descomposición Dantzig-Wolfe e implementación de los llamados *métodos de generación de columnas*, los cuales además permiten incluirlos dentro de un marco de B&B para aumentar el rendimiento del método cuando la formulación tiene un gran número de columnas. Los marcos de trabajo más desarrollados que apoyan la implementación de algoritmos basados en generación de columnas son MINTO [66], ABACUS [46] y COIN/BCP [75] y SCIP [2]. En cada uno de estos marcos de trabajo la implementación surge de formulaciones basadas en una descomposición de Dantzig-Wolfe creada a su vez de una formulación original, más compacta, lo cual puede resultar en complicar la implementación si tal formulación de Dantzig-Wolfe no es tan compacta y sencilla como la formulación original. En la práctica, el uso de métodos de descomposición se ve obstaculizado por una serie de inconvenientes:

- La implementación es difícil, por lo general se requiere del desarrollo de sofisticados códigos personalizados, no resulta genérico.
- La elección de una estrategia algorítmica de descomposición requiere un profundo conocimiento de la teoría y las estrategias son difíciles de comparar

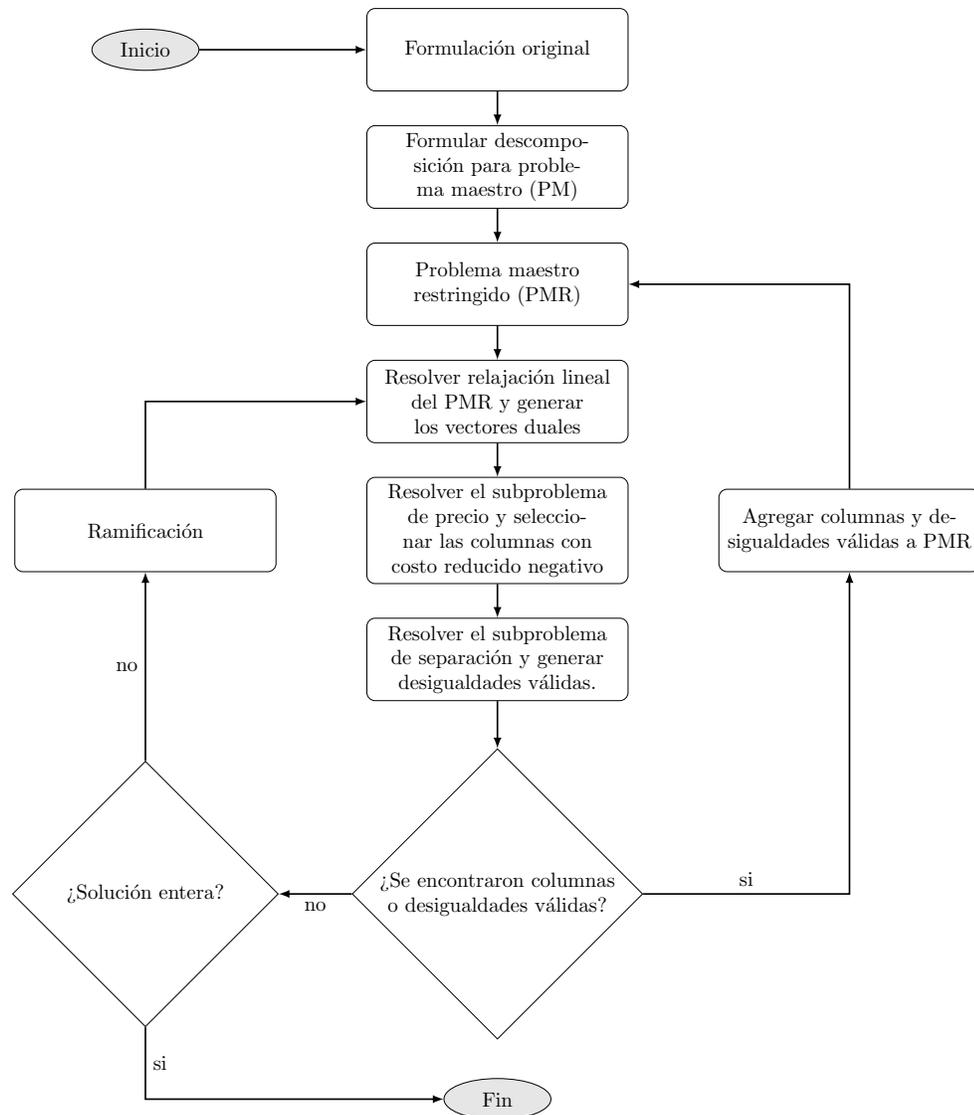


Figura B.3: Diagrama de flujo de BP&C

empíricamente.

- Los modernos y poderosos solucionadores utilizados para resolver programas enteros son difíciles de integrar con los enfoques basados en la descomposición.

DIP (por sus siglas en inglés: *decomposition in integer programming*) es un software creado por Galati y Ralphs [31, 30] que simplifica el proceso de implementación de los métodos de descomposición. DIP provee un completo solucionador

MILP basado en descomposición que se puede utilizar de modo genérico y no requiere una personalización especializada de los componentes del algoritmo basado en las características del problema.

La filosofía de DIP es eliminar la mayor cantidad de carga durante la implementación por parte del usuario, tanto como sea posible, manteniendo las ventajas habituales de una metodología basada en la descomposición. Con DIP, cualquier reformulación requerida se mantiene y realiza internamente, el usuario interactúa sólo con la formulación original compacta. Además, proporciona implementaciones por defecto para todos los componentes del solucionador y pone en práctica una amplia gama de algoritmos basados en descomposición, incluyendo métodos tradicionales como Dantzig-Wolfe, Lagrange y planos cortantes, así como los métodos híbridos con B&B más avanzados, tales como B&C y BP&C.

APÉNDICE C

DEMOSTRACIÓN DE LA COMPLEJIDAD COMPUTACIONAL DEL CpCP

Teorema C.1 *El problema de vértice p -centro capacitado (CpCP) es NP-duro.*

Demostración: El problema debe de ser formulado como un problema de decisión que denotaremos como D-CpCP, por lo cual se debe de formular una instancia genérica del problema.

Instancia: Dado un conjunto V de nodos los cuales representan puntos potenciales de instalaciones y demanda, para cada $i \in V$ se define una capacidad s_i y para cada $j \in V$ una demanda w_j , p es un parámetro que indica el número de instalaciones a ser ubicadas, finalmente se define una distancia entera entre cada par de nodos $i, j \in V$, denotada por d_{ij} .

Definición C.2 *D-CpCP: sea β un entero positivo ¿existe una p -partición X de V tal que*

$$\max_{k \in K} \max_{j \in X_k} \{d_{c(k)j}\} \leq \beta, \quad (\text{C.1})$$

y que además sean satisfechas las restricciones de capacidad y de asignación única?

Primeramente se debe de demostrar que D-CpCP pertenece a la clase NP, para lo cual es necesario probar que dada una p -partición X es posible verificar la respuesta a D-CpCP en tiempo polinomial.

- Decidir si X cumple la condición (C.1), requiere verificar cada partición, para lo cual es necesario invocar la función (2.7), cuyo orden de operaciones es $\mathcal{O}(n^2)$. Verificar la integridad de la capacidad va implícito en esta decisión, dado que la función (2.7) selecciona al nodo verificando esta restricción; si ningún nodo $j \in X_k$ cumple la condición de capacidad, entonces $f(X_k) = \infty$, y por lo tanto se concluye que X es infactible.
- Validar la asignación única de los nodos a una sola instalación requiere verificar la pertenencia de cada nodo a cada partición, lo cual necesita $\mathcal{O}(n^2)$ operaciones.

A partir de lo anterior, se concluye que dar respuesta al D-CpCP, así como comprobar o descartar factibilidad puede verificarse en tiempo polinomial en el tamaño del conjunto V y por lo tanto D-CpCP \in NP. Ahora para demostrar que D-CpCP es NP-completo, es requerido reducir en tiempo polinomial y en espacio logarítmico una instancia de un problema conocido como NP-completo a una instancia de D-CpCP [14, 34]. Para lograr esto nos basamos en el conocido problema p CP, el cual es NP-completo [48] y denotaremos por D- p CP su problema de decisión.

Definición C.3 *D- p CP: sea β un entero positivo ¿existe una p -partición X de V tal que*

$$\max_{k \in K} \max_{j \in X_k} \{d_{c(k)j}\} \leq \beta, \quad (\text{C.2})$$

y que además sean satisfechas las restricciones de asignación única?

En el caso del p CP, la función $c(k)$ que determina el centro de la partición X_k , no contempla la restricción de capacidad. La idea es realizar la transformación D- p CP \propto D-CpCP en tiempo polinomial y probar que la respuesta D-CpCP equivale a la respuesta de D- p CP. Para ello dada a una instancia particular D- p CP, formada de la manera siguiente: un conjunto V de nodos los cuales representan puntos potenciales de instalaciones y de demanda, para cada $j \in V$ una demanda w_j , p es un parámetro que indica el número de instalaciones a ser ubicadas, y una distancia entera entre

cada par de nodos $i, j \in V$, denotada por d_{ij} . Una instancia válida para el $D-C_pCP$, incluye todos los parámetros anteriores y además requiere definir una capacidad para cada nodo $i \in V$, entonces para realizar la transformación $D-pCP \propto D-C_pCP$, se requiere copiar la información de nodos, parámetros y asignar una capacidad a cada nodo, la cual establecemos como $s_i = \infty, i \in V$. El proceso $D-pCP \propto D-C_pCP$ puede realizarse en espacio logarítmico, pues bastará con recorrer la instancia $D-pCP$ de manera secuencial y realizar la copia hacia la nueva instancia para el $D-C_pCP$, agregando los datos de la capacidad.

Argumentaremos la equivalencia de la respuesta entre $D-C_pCP$ y $D-pCP$. Si la respuesta a $D-C_pCP$ es «Sí» con la instancia transformada de $D-pCP$, implica que la p -partición cumple con la condición definida en C.2. La respuesta es claramente válida para el $D-pCP$, ya que la definición C.3 no restringe la capacidad de las instalaciones potenciales, la cual para la instancia transformada se ha definido como infinito. Cuando la respuesta es «No», se argumenta de manera similar, dado que por las características de la instancia, las condiciones de capacidad siempre serán satisfechas, no así, las de asignación única y de distancia menor igual a β .

Con lo anterior, queda demostrado que $D-C_pCP$ es un problema NP-completo y por lo tanto, su versión de optimización es NP-duro. ■

APÉNDICE D

ANÁLISIS DE DESEMPEÑO DE B&B PARA LA SOLUCIÓN DEL $CpCP$

En este apéndice se define un experimento que realiza la ejecución de B&B para la solución del $CpCP$ para el conjunto de instancias clásicas definidas en Biblioteca-OR [11] (A), Galvão-ReVelle [32] (B) y Lorena-Senne [56] (C) (la Sección 5.2.1 describe a detalle las características de cada conjunto) y se realiza además un análisis asintótico en función del tamaño de la instancia las limitantes del método.

D.1 CONDICIONES Y RESULTADOS

Las condiciones experimentales son las siguientes: se usa ILOG CPLEX 12.5 como optimizador de MILP. Se establece para el método un límite de tiempo de 1 hora y un uso de memoria de 4 GB como criterio para finalizar la ejecución. El método se han compilado usando g++ versión 4.4.5 bajo Debian 6.0.8 GNU/Linux Kernel 2.6.32-5, 64 bits. Para el experimento se utilizó un *cluster* AMD Opteron con 2.0 GHz (x16), 32 GB en RAM.

Las Tablas D.1–D.3 presentan los resultados para cada conjunto de instancias. En cada tabla, las dos primeras columnas representan el tamaño de la instancia medido por el número de nodos n y el número de particiones p . «Instancia» es el nombre de la instancia del problema y «Óptimo» indica el valor óptimo de la instancia, valor obtenido por los métodos de Özsoy y Pınar [69] y Albareda-Sambola,

Díaz y Fernández [4]. « f^{lb} » y « f^{ub} » es la mejor cota inferior y superior encontrada por B&B respectivamente. La columna «Tiempo (seg)» muestra el tiempo en segundos y «Desviación (%)» expresa el porcentaje de desviación relativa entre las cotas inferior y superior. A continuación se discuten los resultados de cada una de las tablas.

Tabla D.1: Resultado de B&B en la solución del conjunto de instancias A

n	p	Instancia	Óptimo	f^{lb}	f^{ub}	Desviación (%)	Tiempo (seg)	Memoria (B)
50	5	cpmp01	29	29	29	0.00	4.06	4.21E+7
		cpmp02	33	33	33	0.00	1.18	3.68E+7
		cpmp03	26	26	26	0.00	2.75	1.59E+7
		cpmp04	32	32	32	0.00	2.69	2.36E+7
		cpmp05	29	29	29	0.00	2.88	3.78E+7
		cpmp06	31	31	31	0.00	3.04	2.08E+7
		cpmp07	30	30	30	0.00	2.24	1.90E+7
		cpmp08	31	31	31	0.00	3.32	4.01E+7
		cpmp09	28	28	28	0.00	2.85	3.93E+7
		cpmp10	32	32	32	0.00	3.23	2.71E+7
Promedio						0.00	2.82	3.03E+7
100	10	cpmp11	19	19	19	0.00	24.59	1.15E+8
		cpmp12	20	20	20	0.00	24.96	1.13E+8
		cpmp13	20	20	20	0.00	21.15	5.34E+7
		cpmp14	20	20	20	0.00	18.89	1.19E+8
		cpmp15	21	21	21	0.00	50.29	1.01E+8
		cpmp16	20	20	20	0.00	21.61	9.18E+7
		cpmp17	22	22	22	0.00	3322.99	6.30E+7
		cpmp18	21	21	21	0.00	104.06	8.39E+7
		cpmp19	21	21	21	0.00	41.34	5.08E+7
		cpmp20	21	21	21	0.00	272.82	6.68E+7
Promedio						0.00	390.27	8.58E+7
Promedio Total						0.00	196.55	5.80E+7

Respecto al conjunto de instancias A (Tabla D.1), el método es eficiente para el pequeño grupo de instancias (tamaño 50×5). Para el grupo de mayor tamaño (tamaño 100×10), encontramos un par de instancias para las cuales resultan más difíciles de resolver; sin embargo el método encuentra el óptimo para todas las instancias de este conjunto.

Para el conjunto de instancias B (Tabla D.2), conforme el tamaño de la instancia crece, B&B no encuentra el óptimo en la mayoría de las instancias, incrementando el tiempo de computo y reportando un valor elevado de desviación relativa entre las cotas. Finalmente, el conjunto de instancias C (Tabla D.3), B&B no encuentra el

óptimo en ninguna instancia, reportando a su vez un valor bastante elevado de desviación entre cotas (superior al 100 %). Cabe señalar que la mayoría de las instancias de los conjuntos B y C, pararon su ejecución por rebasar el límite de memoria permitido.

Tabla D.2: Resultado de B&B en la solución del conjunto de instancias B

n	p	Instancia	Óptimo	f^{lb}	f^{ub}	Desviación (%)	Tiempo (seg)	Memoria (B)
100	5	G1	94	93	93	0.00	185.71	6.90E+7
100	5	G2	94	85	97	14.12	207.11	8.09E+8
100	10	G3	83	83	83	0.00	589.14	1.09E+8
100	10	G4	84	55	92	67.27	88.72	7.65E+8
150	10	G5	95	77	103	33.77	266.37	8.98E+8
150	10	G6	96	77	103	33.77	167.81	9.21E+8
150	15	G7	89	67	99	47.76	185.52	8.65E+8
150	15	G8	89	67	97	44.78	117.53	8.47E+8
Promedio Total						30.18	225.99	6.60E+8

Tabla D.3: Resultado de B&B en la solución del conjunto de instancias C

n	p	Instancia	Óptimo	f^{lb}	f^{ub}	Desviación (%)	Tiempo (seg)	Memoria (B)
100	10	SJC1	364	364	364	0.00	88.25	1.51E+8
200	15	SJC2	304	236	470	99.15	188.79	8.91E+8
300	25	SJC3a	278	242	390	61.16	1050.20	1.22E+9
300	30	SJC3b	253	224	392	75.00	882.60	1.15E+9
402	30	SJC4a	284	185	497	168.65	1192.97	3.12E+9
402	40	SJC4b	239	160	398	148.75	880.65	3.13E+9
Promedio Total						92.12	713.91	1.61E+9

D.2 CONCLUSIONES

La Figura D.1 muestra el análisis asintótico de B&B para la solución del C_pCP . Tal como se observa el consumo de recurso crece de manera acelerada respecto al tamaño de la instancia. El tiempo promedio respecto al tamaño tiene un brusco cambio de 5 a 100 nodos, y parece mantenerse en valores similares en los rangos siguientes, sin embargo cabe recordar que la mayoría de las instancias finalizaron por exceder el consumo de memoria permitido. Las estadísticas de memoria indican el máximo tamaño del conjunto residente utilizado en bytes [55], esto es, el máximo número

de bytes de la memoria física que cada proceso usa simultáneamente. Tal como se observa la cantidad de memoria requerida por el B&B incrementa considerablemente conforme crece el tamaño de la instancia.

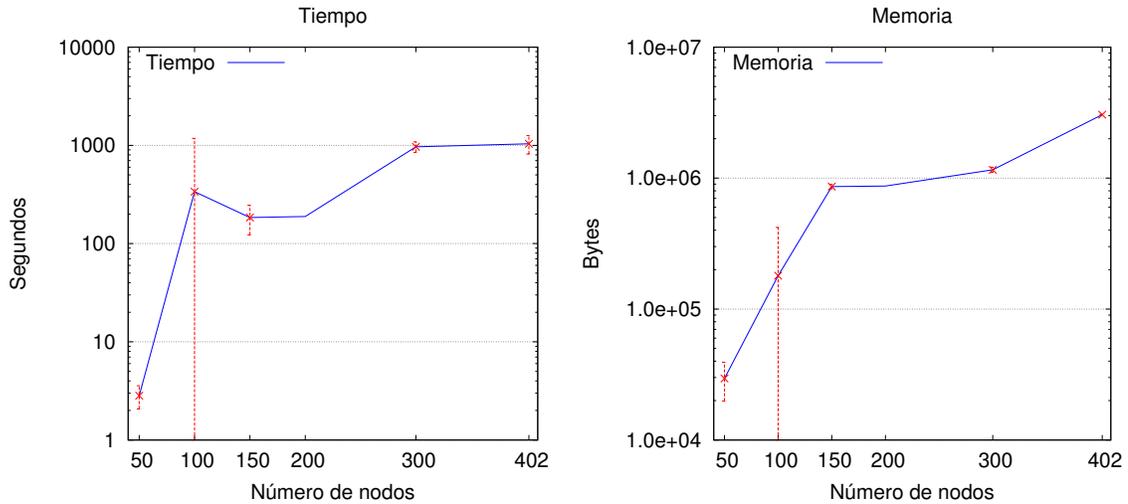


Figura D.1: Comparación asintótica en términos de tiempo de ejecución y uso de memoria de B&B. Los ejes verticales se encuentran en escala logarítmica. La barra vertical expresa la desviación estándar.

Con lo anterior, se puede concluir que B&B sólo resulta eficiente para la solución del C_p CP cuando el tamaño de la instancia es menor a 100 nodos, y mantiene un valor de p proporcional a este tamaño. Para instancias de mayor tamaño, el método no asegura un consumo de recursos razonable que aporte una solución factible al problema.

APÉNDICE E

GRÁFICAS DE ANÁLISIS ESTADÍSTICO

E.1 ANÁLISIS ESTADÍSTICO DE DESVIACIÓN RELATIVA

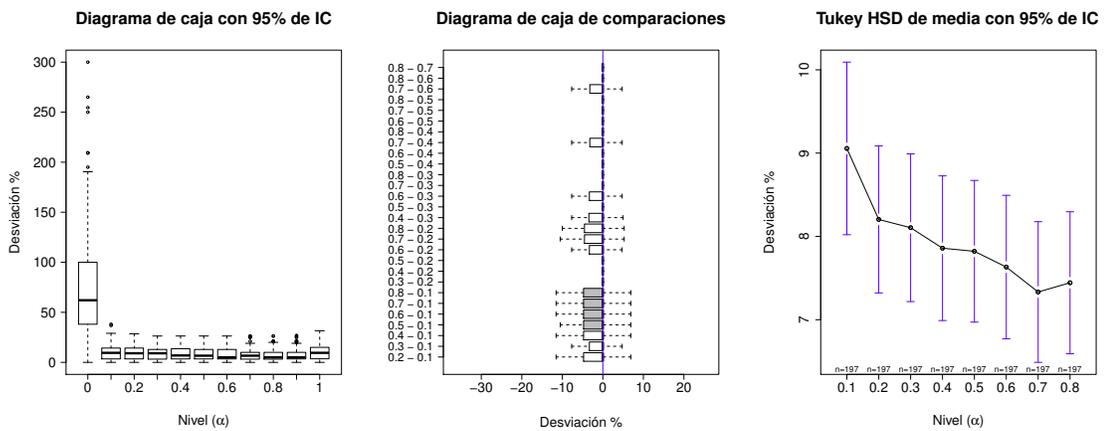


Figura E.1: Análisis de la desviación porcentual en el conjunto de instancias A.

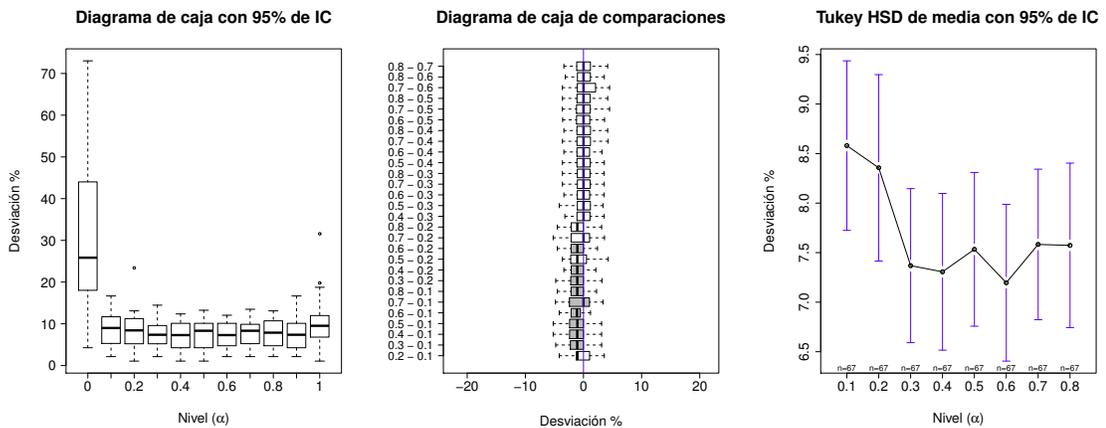


Figura E.2: Análisis de la desviación porcentual en el conjunto de instancias B.

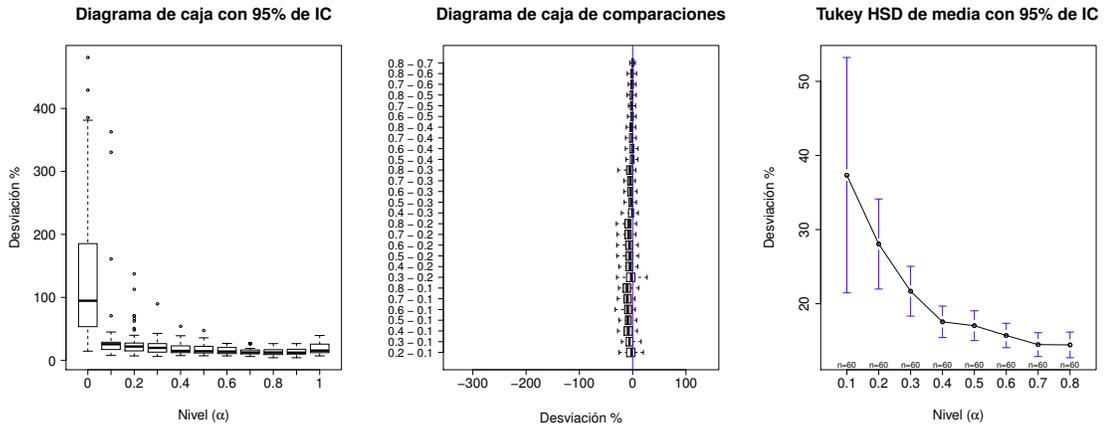


Figura E.3: Análisis de la desviación porcentual en el conjunto de instancias C.

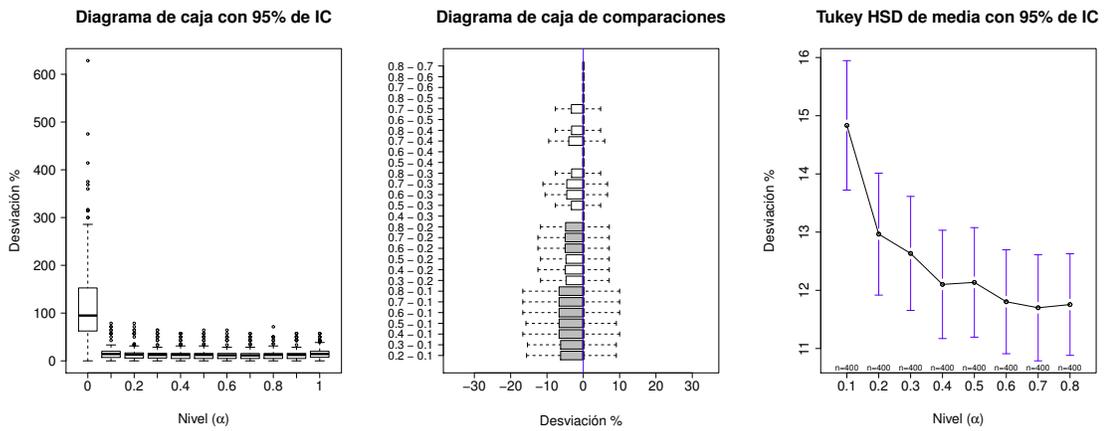


Figura E.4: Análisis de la desviación porcentual en el conjunto de instancias D- α .

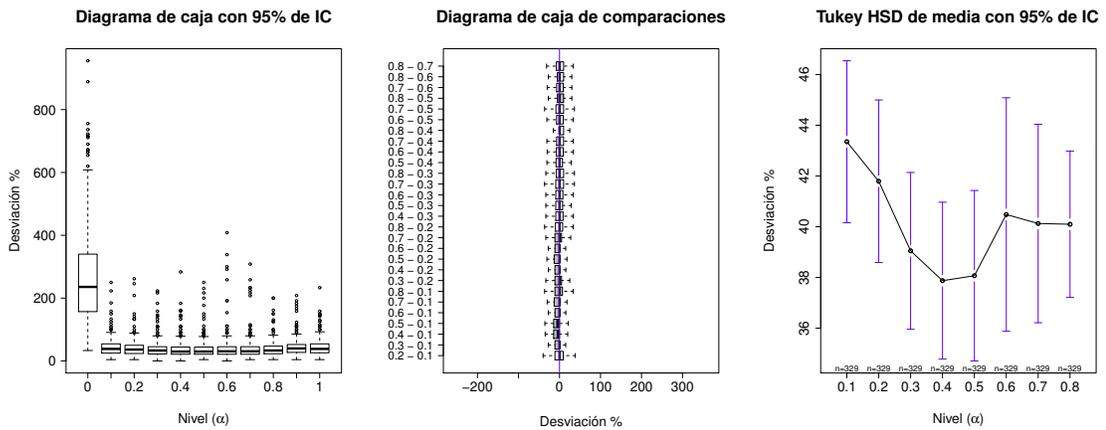


Figura E.5: Análisis de la desviación porcentual en el conjunto de instancias D- β .

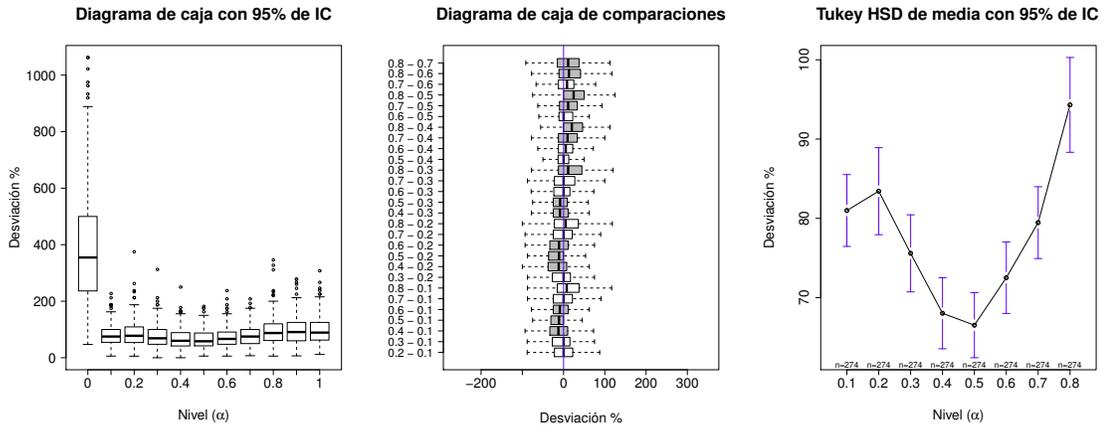


Figura E.6: Análisis de la desviación porcentual en el conjunto de instancias D- γ .

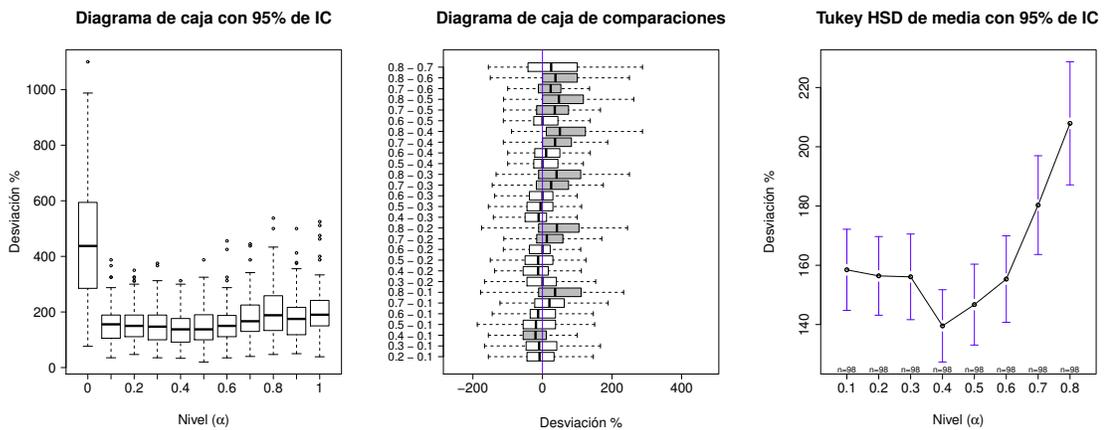


Figura E.7: Análisis de la desviación porcentual en el conjunto de instancias D- δ .

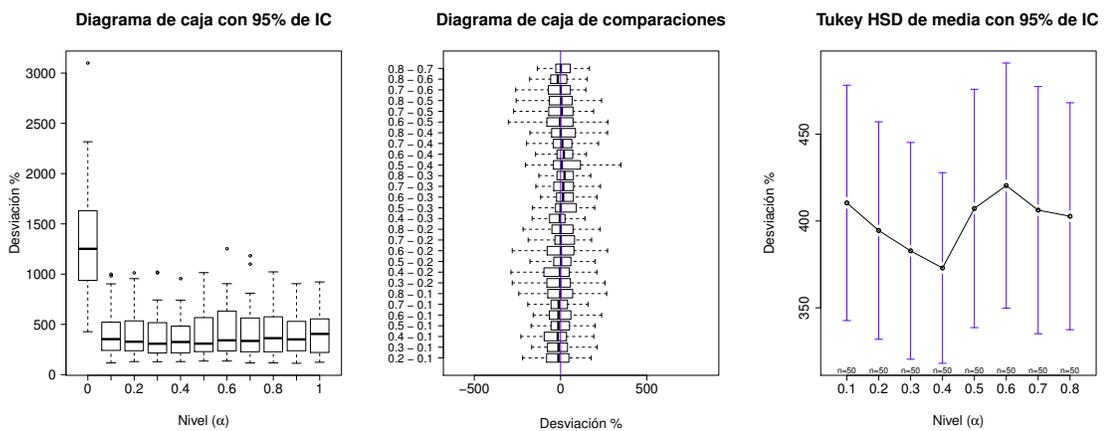


Figura E.8: Análisis de la desviación porcentual en el conjunto de instancias E

E.2 ANÁLISIS DEL TIEMPO

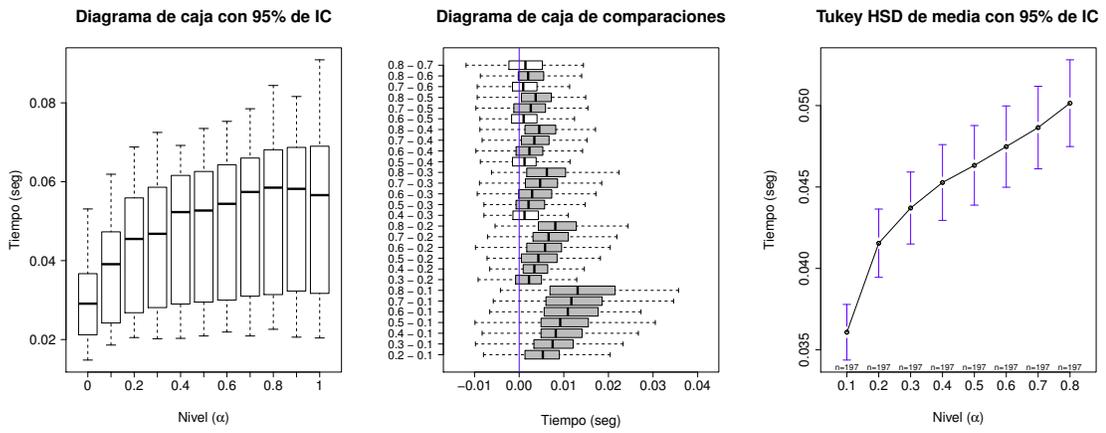


Figura E.9: Análisis del tiempo para el conjunto de instancias A.

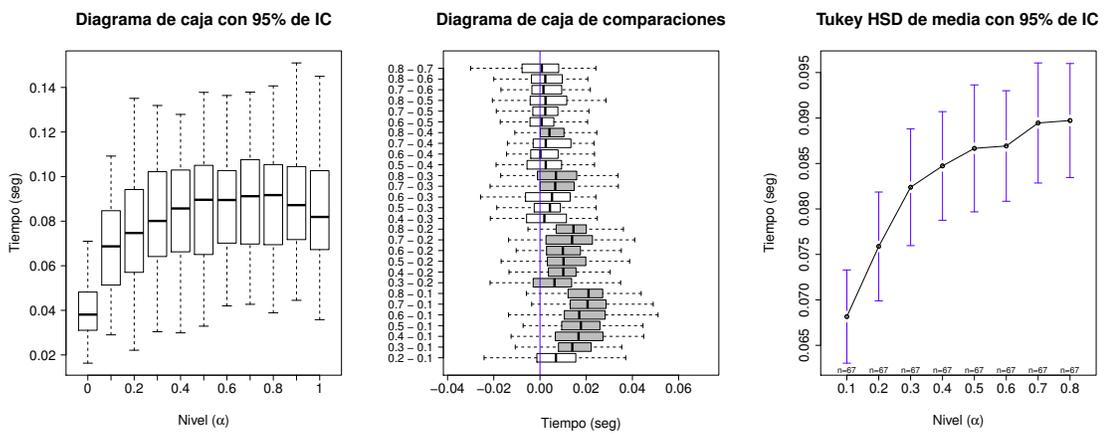


Figura E.10: Análisis del tiempo para el conjunto de instancias B.

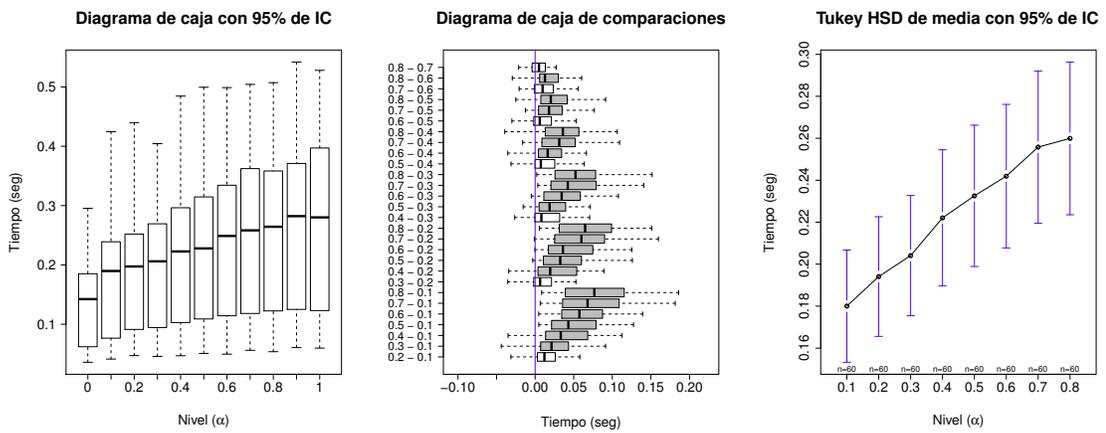


Figura E.11: Análisis del tiempo para el conjunto de instancias C.

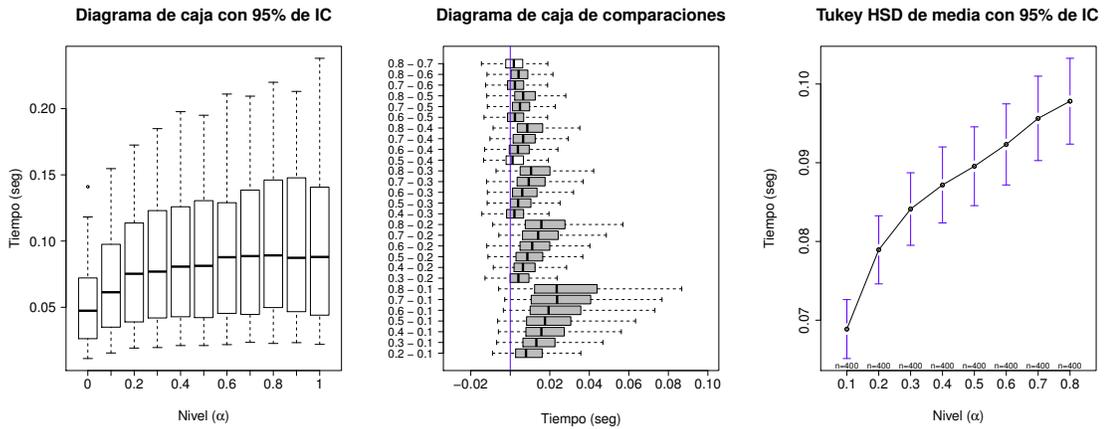


Figura E.12: Análisis del tiempo para el conjunto de instancias D- α .

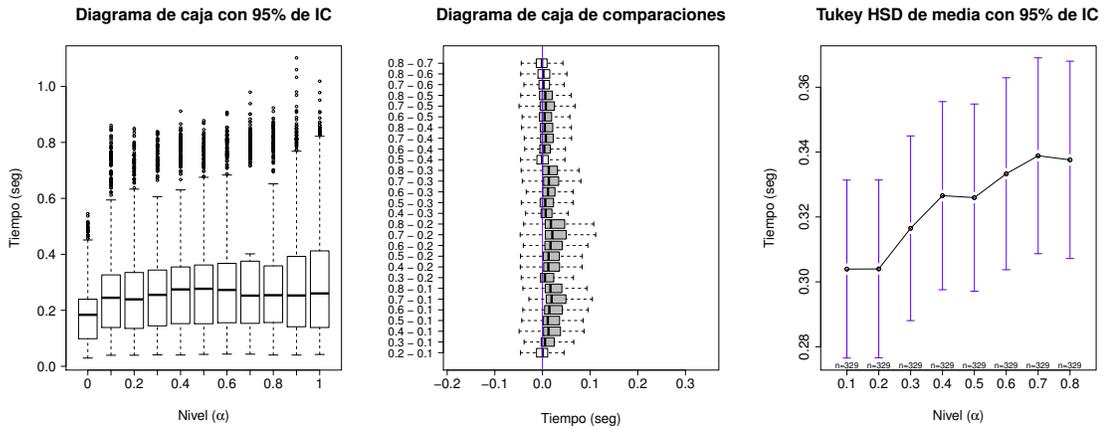


Figura E.13: Análisis del tiempo para el conjunto de instancias D- β .

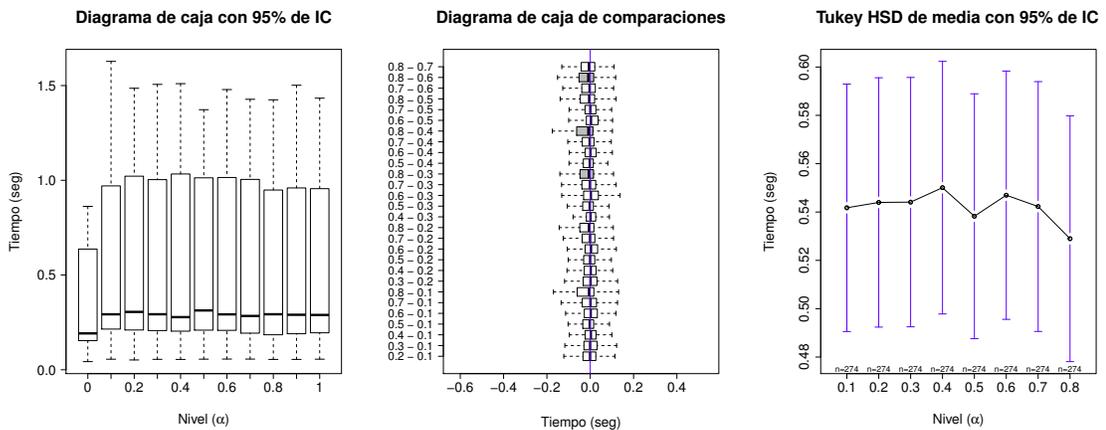


Figura E.14: Análisis del tiempo para el conjunto de instancias D- γ .

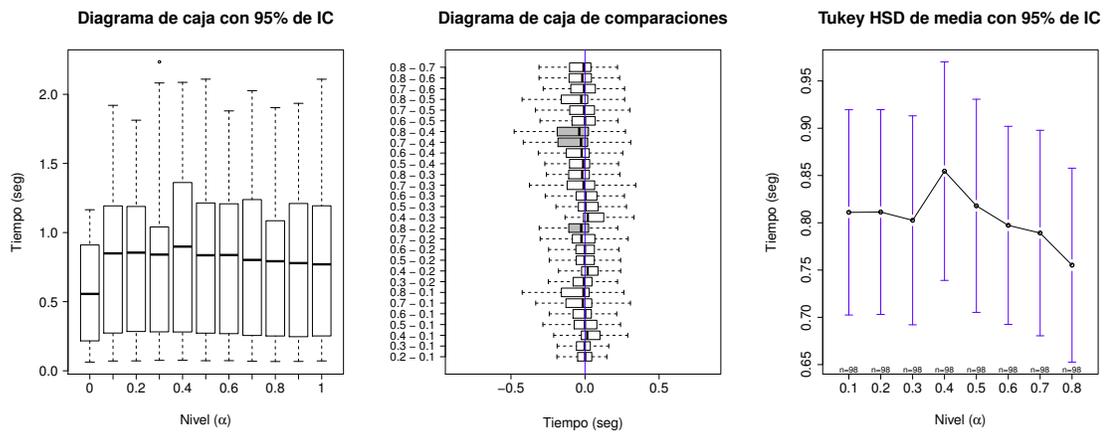


Figura E.15: Análisis del tiempo para el conjunto de instancias D- δ .

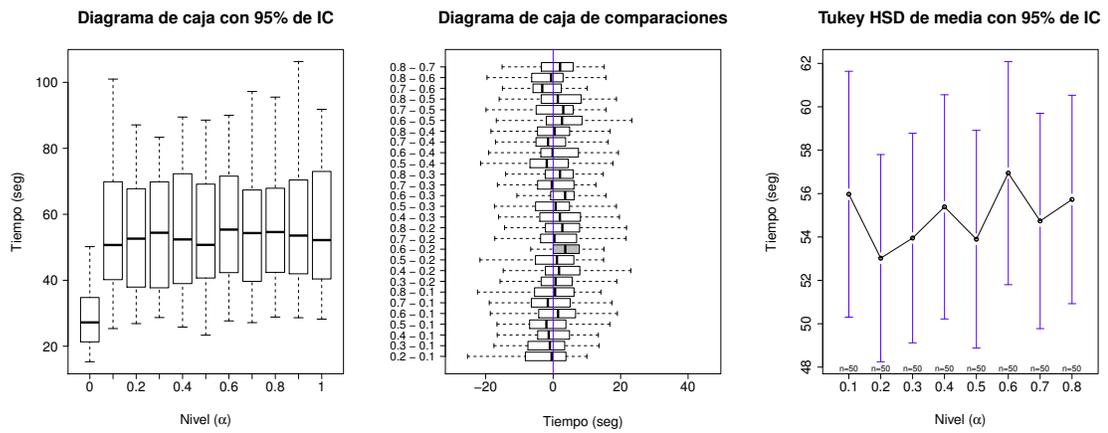


Figura E.16: Análisis del tiempo para el conjunto de instancias E.

BIBLIOGRAFÍA

- [1] H. Abdi y L. J. Williams. Honestly significant difference (HSD) test. En N. J. Salkind, D. M. Dougherty y B. Frey, editores, *Encyclopedia of Research Design*, páginas 583–585. Sage, Thousand Oaks, EUA, 2010.
- [2] T. Achterberg. Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.
- [3] R. K. Ahuja, T. L. Magnanti y J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Upper Saddle River, EUA, 1993.
- [4] M. Albareda-Sambola, J. A. Díaz y E. Fernández. Lagrangean duals and exact solution to the capacitated p -center problem. *European Journal of Operational Research*, 201(1):71–81, 2010.
- [5] A. Amiri. A probabilistic greedy algorithm for channel assignment in cellular radio networks. *IEEE Transactions on Communications*, 58(11):3286–3295, 2010.
- [6] J. Bar-Ilan, G. Kortsarz y D. Peleg. How to allocate network centers. *Journal of Algorithms*, 15(3):385–415, 1993.
- [7] C. Barnhart, C. A. Hane y P. H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

-
- [8] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh y P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- [9] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende y W. R. Stewart. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1):9–32, 1995.
- [10] J. Beasley. Lagrangean relaxation. En C. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Optimization*, capítulo 6. Wiley, New York, EUA, 1993.
- [11] J. E. Beasley. OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
- [12] B. Ç. Tansel. Discrete center problems. En H. A. Eiselt y V. Marianov, editores, *Foundations of Location Analysis*, volumen 155 de *International Series in Operations Research & Management Science*, capítulo 5, páginas 79–106. Springer, New York, EUA, 2011.
- [13] A. Ceselli y G. Righini. A branch-and-price algorithm for the capacitated p -median problem. *Networks*, 45(3):125–142, 2005.
- [14] S. A. Cook. The complexity of theorem-proving procedures. En *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, páginas 151–158, ACM. New York, EUA, 1971.
- [15] G. B. Dantzig y P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.
- [16] M. Poggi de Aragão y E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price algorithms. En *Proceedings of Mathematical Programming in Rio: A Conference in Honour of Nelson Maculan*, páginas 56–61, Río de Janeiro, Brasil, Noviembre, 2003.

- [17] J. Desrosiers y M. E. Lübbecke. A primer in column generation. En M. M. Solomon, editor, *Column Generation*, capítulo 1, páginas 1–29. Springer, Heidelberg, Alemania, 2005.
- [18] J. Desrosiers y M. E. Lübbecke. Branch-price-and-cut algorithms. En J. J. Cochran, L. A. Cox, P. Keskinocak, J. P. Kharoufeh y J. C. Smith, editores, *Wiley Encyclopedia of Operations Research and Management Science*, volumen 8. Wiley, New York, EUA, 2010.
- [19] A. Díaz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato y F. T. Tseng. *Optimización Heurística y Redes Neuronales*. Paraninfo, Madrid, España, 1996.
- [20] J. A. Díaz y E. Fernández. A branch-and-price for the source algorithm single plant location problem capacitated. *Journal of the Operational Research Society*, 53(7):728–740, 2002.
- [21] M. Dyer y A. Frieze. A simple heuristic for the p -center problem. *Operations Research Letters*, 3(6):285–288, 1985.
- [22] S. Eilon. More against optimization. *Omega*, 5(1):627–633, 1977.
- [23] M. G. Elizondo-Amaya, R. Z. Ríos-Mercado y J. A. Díaz. A dual bounding scheme for a territory design problem. *Computers & Operations Research*, 44:193–205, 2014.
- [24] S. Elloumi, M. Labbé y Y. Pochet. A new formulation and resolution method for the p -center problem. *INFORMS Journal on Computing*, 16(1):84–94, 2004.
- [25] L. Fanjul-Peyro y R. Ruiz. Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research*, 207(1):55–69, 2010.
- [26] R. Z. Farahani y M. Hekmatfar, editores. *Facility Location: Concepts, Models, Algorithms and Case Studies*. Physica-Verlag, Heidelberg, Alemania, 2009.

-
- [27] M. L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 50(12):1861–1871, 2004.
- [28] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11(1):86–92, 1940.
- [29] M. V. Galati. *Decomposition Methods for Integer Linear Programming*. Tesis de doctorado, Lehigh University, Bethlehem, EUA, 2009.
- [30] M. V. Galati y T. K. Ralphs. DIP: A framework for decomposition in integer programming. Reporte técnico, COR@L Laboratory, Lehigh University, Bethlehem, EUA, 2013.
- [31] M. V. Galati, T. K. Ralphs y J. Wang. Computational experience with generic decomposition using the DIP framework. Reporte Técnico 12T-014, COR@L Laboratory, Lehigh University, Bethlehem, EUA, 2012.
- [32] R. D. Galvão y C. ReVelle. A lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research*, 88(1):114–123, 1996.
- [33] G. Gamrath. *Generic Branch-Cut-and-Price*. Tesis de doctorado, Technische Universität Berlin, Berlin, Alemania, 2010.
- [34] M. Garey y D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, EUA, 1979.
- [35] M. Gendreau y J.-Y. Potvin. *Handbook of Metaheuristics*, volumen 146 de *International Series in Operations Research & Management Science*, 2a edición. Springer, New York, EUA, 2010.
- [36] A. M. Geoffrion. Lagrangian relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.
- [37] F. Glover y M. Laguna. *Tabu Search*. Kluwer, Boston, EUA, 1997.

- [38] S. L. Hakimi. Optimum location of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.
- [39] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.
- [40] P. Hansen y N. Mladenović. An introduction to variable neighborhood search. En S. Voss, S. Martello, I. H. Osman y C. Roucairol, editores, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, páginas 433–458. Kluwer, Boston, EUA, 1999.
- [41] P. Hansen y N. Mladenović. J-Means: A new local search heuristic for minimum sum of squares clustering. *Pattern Recognition*, 34(2):405–413, 2001.
- [42] P. Hansen y N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.
- [43] M. Haouari y J. S. Chaouachi. A probabilistic greedy search algorithm for combinatorial optimization with application to the set covering problem. *Journal of the Operational Research Society*, 53(7):792–799, 2002.
- [44] M. Jaeger y J. Goldberg. A polynomial algorithm for the equal capacity p -center problem on trees. *Transportation Science*, 28(2):167–75, 1994.
- [45] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.
- [46] M. Jünger y S. Thienel. Introduction to ABACUS—a branch-and-cut system. *Operations Research Letters*, 22(1):83–95, 1998.
- [47] J. Kalcsics, S. Nickel y M. Schröder. Toward a unified territorial design approach: Applications, algorithms, and GIS integration. *Top*, 13(1):1–56, 2005.
- [48] O. Kariv y S. L. Hakimi. An algorithmic approach to network location problems, I: the p -centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

-
- [49] J. P. Kelly y I. Osman. *Meta-Heuristics: Theory and Applications*. Kluwer, Boston, EUA, 1996.
- [50] S. Khuller y Y. J. Sussmann. The capacitated K-center problem. *SIAM Journal on Discrete Mathematics*, 13(3):403–418, 2000.
- [51] A. Klose y S. Görtz. A branch-and-price algorithm for the capacitated facility location problem. *European Journal of Operational Research*, 179(3):1109–1125, 2007.
- [52] W. H. Kruskal y W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.
- [53] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Saunders College Publishing, Fort Worth, EUA, 1976.
- [54] C. Lemarechal. Lagrangean relaxation. En M. Jünger y D. Naddef, editores, *Computational Combinatorial Optimization*. Springer, New York, EUA, 2001.
- [55] S. Loosemore, R. M. Stallman, R. McGrath, A. Oram y U. Drepper. *The GNU C Library Reference Manual: for version 2.17*. Free Software Foundation, 2012.
- [56] L. A. N. Lorena y E. L. F. Senne. A column generation approach to capacitated p -median problems. *Computers & Operations Research*, 31(6):863–876, 2004.
- [57] H. R. Lourenço, O. Martin y T. Stützle. Iterated local search. En F. Glover y G. A. Kochenberger, editores, *Handbook of Metaheuristics*, páginas 321–353. Kluwer, Norwell, EUA, 2002.
- [58] H. Marchand y L. A. Wolsey. Aggregation and mixed integer rounding to solve MIPs. *Operations Research*, 49(3):363–371, 2001.
- [59] A. A. Márkov. Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain. En R. Howard, editor, *Dynamic Probabilistic Systems: Markov Models*, volumen I, apéndice B, páginas 552–577. Wiley, New York, EUA, 1971.

- [60] R. Martí. Algoritmos heurísticos en optimización combinatoria. Reporte técnico, Departamento de Estadística e Investigación Operativa, Facultad de Ciencias Matemáticas, Universidad de Valencia, Valencia, España, 2003.
- [61] A. Martin y R. Weismantel. Contributions to general mixed integer knapsack problems. Reporte Técnico 38, Zuse-Institut Berlin, Berlin, Alemania, 1997.
- [62] A. X. Martins, C. Duhamel, M. C. Souza, R. R. Saldanha y P. Mahey. A VND-ILS heuristic to solve the RWA problem. En J. Pahl, T. Reiners y S. Voß, editores, *Network Optimization*, volumen 6701 de *Lecture Notes in Computer Science*, páginas 577–582. Springer, Berlin, Alemania, 2011.
- [63] N. Mladenović, M. Labbé y P. Hansen. Solving the p -center problem with tabu search and variable neighborhood search. *Networks*, 42(1):48–64, 2003.
- [64] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, EUA, 2005.
- [65] M. G. Narciso y L. A. N. Lorena. Lagrangean/surrogate relaxation for generalized assignment problems. *European Journal of Operational Research*, 114(1):165–177, 1999.
- [66] G. L. Nemhauser, M. W. P. Savelsbergh y G. C. Sigismondi. MINTO, a Mixed INTeger Optimizer. *Operations Research Letters*, 15(1):47–58, 1994.
- [67] G. L. Nemhauser y L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, EUA, 1988.
- [68] M. O’Sullivan, Q. Lim, C. Walker, I. Dunning y S. Mitchell. Dippy — a simplified interface for advanced mixed-integer programming. Reporte Técnico 685, Faculty of Engineering, University of Auckland, Auckland, Nueva Zelanda, 2012.
- [69] F. A. Özsoy y M. Ç. Pinar. An exact algorithm for the capacitated vertex p -center problem. *Computers & Operations Research*, 33(5):1420–1436, 2006.

- [70] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, EUA, 1994.
- [71] C. H. Papadimitriou y K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, EUA, 1982.
- [72] R. Qu y. Xu y G. Kendall. A variable neighborhood descent search algorithm for delay-constrained least-cost multicast routing. En T. Stützle, editor, *Learning and Intelligent Optimization*, volumen 5851 de *Lecture Notes in Computer Science*, páginas 15–29. Springer-Verlag, Heidelberg, Alemania, 2009.
- [73] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.
- [74] T. K. Ralphs y M. V. Galati. Decomposition in integer linear programming. Reporte técnico, Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, EUA, 2006.
- [75] T. K. Ralphs y L. Ladányi. *COIN/BCP User's Manual*. Department of Industrial and Manufacturing Systems Engineering, Lehigh University, Bethlehem, EUA, 2001.
- [76] T. K. Ralphs, L. Ladányi y M. J. Saltzman. Parallel branch, cut, and price for large-scale discrete optimization. *Mathematical Programming*, 98(1–3):253–280, 2003.
- [77] G. Reinelt. *The Traveling Salesman: Computational Solutions for TSP Applications*, volumen 840 de *Lecture Notes in Computer Science*. Springer, Heidelberg, Alemania, 1994.
- [78] C. S. ReVelle y H. A. Eiselt. Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19, 2005.
- [79] C. C. Ribeiro, D. Aloise, T. F. Noronha, C. Rocha y S. Urrutia. An efficient implementation of a VNS/ILS heuristic for a real-life car sequencing problem. *European Journal of Operational Research*, 191(3):596–611, 2008.

-
- [80] R. Z. Ríos-Mercado y E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.
- [81] N. Rodrigues-Lima. A column generation technique for a crisis deployment planning problem. Tesis de maestría, Naval Postgraduate School, Monterey, California, EUA, 1988.
- [82] R. Ruiz y T. Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049, 2007.
- [83] R. Ruiz y T. Stützle. An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3):1143–1159, 2008.
- [84] M. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6):831–841, 1997.
- [85] M. P. Scaparra, S. Pallottino y M. G. Scutellà. Large-scale local search heuristics for the capacitated vertex p -center problem. *Networks*, 43(4):241–255, 2004.
- [86] E. L. F. Senne, L. A. N. Lorena y M. A. Pereira. A branch-and-price approach to p -median location problems. *Computers & Operations Research*, 32(6):1655–1664, 2005.
- [87] A. Subramanian, L. M. A. Drummond, C. Bentes, L. S. Ochi y R. Farias. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911, 2010.
- [88] T. Urlings, R. Ruiz y T. Stützle. Shifting representation search for hybrid flexible flowline problems. *European Journal of Operational Research*, 207(2):1086–1095, 2010.

-
- [89] F. Vanderbeck. On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128, 2000.
- [90] L. Vargas-Suárez, R. Z. Ríos-Mercado y F. López. usando GRASP para resolver un problema de definición de territorios de atención comercial. En M. G. Arenas, F. Herrera, M. Lozano, J. J. Merelo, G. Romero y A. M. Sánchez, editores, *Proceedings of the IV Spanish Conference on Metaheuristics, Evolutionary and Bioinspired Algorithms*, páginas 609–617, Granada, España, 2005.
- [91] L. A. Wolsey. *Integer Programming*, volumen 52. Wiley, New York, EUA, 1998.

ÍNDICE ALFABÉTICO

- análisis
 - algorítmico, 27
 - asintótico, 27, 91
 - de componentes, 62
 - de desempeño, 89
 - estadístico, 43, 93
 - de Friedman, 45
 - de Kruskal-Wallis, 44
 - de Tukey, 45
- complejidad
 - computacional, 12, 72, 86
 - del algoritmo, 27, 72
- descomposición, 75, 76
 - de Benders, 75
 - de Dantzig-Wolfe, 36, 79
 - de Lagrange, 79
 - de planos cortantes, 80
- diseño
 - experimental, 40, 43, 63
 - territorial, 4
- distancia, 68
 - euclidiana, 43, 68
- función
 - crecimiento exponencial, 71
 - crecimiento logarítmico, 71
 - de dispersión, 69
 - de probabilidad, 17, 20
 - de transición, 29
 - minimax, 11
 - objetivo, 10
 - voraz, 17
- grafo, 4, 66
 - conexo, 67
 - dirigido, 66
 - multigrafo, 66
 - plano, 66
 - ponderado, 68
- instancia, 41, 71, 86, 87
 - de prueba, 41, 69
- método de solución
 - exacto
 - Dantzig-Wolfe, 79
 - generación de columnas y cortes, 81
 - ramificación y acotamiento, 89
 - heurístico, 73
 - hibridación, 15

IGLS, 14
ILS, 15
VND, 15
VNS, 15
voraz probabilístico, 14

problema

de precio, 37
combinatorio, 71, 73
de acotamiento, 80
de decisión, 72, 76, 86
de diseño territorial, 4
de generación de columnas, 35, 79
de localización, 1, 2
de optimización, 69, 70, 72, 79
de precio, 81, 82
de separación, 76, 80, 82
ILP, 76
intratable, 72
maestro, 36, 79, 80, 82
 restringido, 80
MILP, 9, 70, 71
sin solución, 72

FICHA AUTOBIOGRÁFICA

Dagoberto Ramón Quevedo Orozco

Candidato para el grado de Maestría en Ciencias en Ingeniería de Sistemas

Facultad de Ingeniería Mecánica y Eléctrica

Universidad Autónoma de Nuevo León

Tesis:

OPTIMIZACIÓN DEL PROBLEMA DEL p -CENTRO CAPACITADO

Nació el 23 de Noviembre de 1986 en Acaponeta, Nayarit. Tercer hijo del Sr. Dagoberto Quevedo Tejeda y la Sra. Ana Elsa Orozco Sánchez. Graduado del Instituto Tecnológico de Tepic (2005 – 2010) como Ingeniero en Sistemas Computacionales. Laboró en el sector gobierno en Servicios de Salud de Nayarit de 2008 a 2011. Inicia sus estudios de Maestría en Ciencias en Ingeniería de Sistemas en la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León en Enero del 2012.