Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirector de Estudios de Posgrado

Estudio de un algoritmo heurístico híbrido basado en la teoría de campo medio aunado a una búsqueda local para el problema de la mochila cuadrática

por

Juan Antonio Banda Moreno

en opción al grado de

Maestría en Ciencias

en Ingeniería de Sistemas

San Nicolás de los Garza, Nuevo León mayo 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirector de Estudios de Posgrado

Estudio de un algoritmo heurístico híbrido basado en la teoría de campo medio aunado a una búsqueda local para el problema de la mochila cuadrática

por

Juan Antonio Banda Moreno

en opción al grado de

Maestría en Ciencias

en Ingeniería de Sistemas

San Nicolás de los Garza, Nuevo León mayo 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirector de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Estudio de un algoritmo heurístico híbrido basado en la teoría de campo medio aunado a una búsqueda local para el problema de la mochila cuadrática», realizada por el alumno Juan Antonio Banda Moreno, con número de matrícula 1407510, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. José Arturo Berrones Santos
Asesor

Dr. Igor S. Litvinchev
Revisor

Dr. Francisco Almaguer Martínez
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, mayo 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirector de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Estudio de un algoritmo heurístico híbrido basado en la teoría de campo medio aunado a una búsqueda local para el problema de la mochila cuadrática», realizada por el alumno Juan Antonio Banda Moreno, con número de matrícula 1407510, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. José Arturo Berrones Santos

Asesor

Dr. Igor S. Litvinchev

Revisor

Dr. Francisco Almaguer Martínez

Revisor

Vo. Bo.

Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, mayo 2014

A mis padres.

Índice general

Agradecimientos xii

Resumen xiii

1. Introducción 1

1.1. Descripción del Problema 4

1.2. Motivación y Justificación 4

2. Marco Teórico 6

2.1. Método voraz (greedy) 9

2.2. Búsqueda local 11

2.3. Inferencia variacional 14

2.3.1. Campo medio 14

2.3.2. Campo medio variacional 14

2.4. Condiciones de Karush-Kuhn-Tucker 17

3. Caso de Estudio 19

3.1. Problema de la Mochila Cuadrática 20

3.1.1. Aplicaciones 21

4. Métodos de Solución 22

4.1. Algoritmo campo medio 23

4.2. Algoritmo campo medio mejorado 24

4.3. Algoritmo híbrido campo medio aunado a una búsqueda local simple 26

4.4. Algoritmo híbrido voraz aunado a una búsqueda local simple 27

5. Experimentación 28

6. Conclusiones y trabajo futuro 35

A. Apéndice 36

Ficha autobiográfica 45

Índice de figuras

2.1. Representación gráfica de un problema de optimización mediante un grafo. 15

3.1. Grafo. 21

5.1. Comparativo de GAP promedio entre CMBL y VBL en casos de tamaño 200. 29

5.2. Comparativo del tiempo promedio de ejecución entre CMBL y VBL en casos de tamaño 200. 29

5.3. Comparativo del mejor porcentaje del valor de la función objetivo entre CMBL y VBL, en casos de tamaño 1000 con una densidad del 100 %. 30

5.4. Comparativo del tiempo promedio de ejecución entre CMVBL y VBL, en casos de tamaño 1000. 31

5.5. Comparativo del mejor porcentaje del valor de la función objetivo entre CMBL y VBL, en casos de tamaño 2000 con una densidad del 100 % 32

5.6. Comparativo del tiempo promedio de ejecución entre CMBL y VBL, en casos de tamaño 2000. 32

5.7. Comparativo del mejor porcentaje del valor de la función objetivo entre CM y Voraz, en casos de tamaño 5000 con una densidad del 100 % 33

5.8. Comparativo del mejor porcentaje del valor de la función objetivo entre CM y Voraz, en casos de tamaño 10000 con una densidad del 100 % 34

5.9. Comparativo del mejor porcentaje del valor de la función objetivo entre CM y Voraz, en casos de tamaño 20000 con una densidad del 100 % 34

Índice de tablas

4.1. Probabilidad de ocurrencia de las variables de decisión 25

A.1. Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 25 %. 36

A.2. Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 50 %. 37

A.3. Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 75 %.. 37

A.4. Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 100 %. 38

A.5. Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 1000 con una densidad del 100 %. 38

A.6. Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 2000 con una densidad del 100 %. 39

A.7. Resultados experimentales obtenidos de CM y Voraz, en tamaños de casos 5000 con una densidad del 100 %. 39

A.8. Resultados experimentales obtenidos de CM y Voraz, en tamaños de casos 10000 con una densidad del 100 %. 40

A.9. Resultados experimentales obtenidos de CM y Voraz, en tamaños de casos 20000 con una densidad del 100 %. 40

Agradecimientos

Primeramente quiero agradecer a mis padres por su comprensión y apoyo, sobre todo por creer siempre en mí. También, deseo agradecer a la Universidad Autónoma de Nuevo León (U.A.N.L) la oportunidad que me brindó para realizar mis estudios. A la Facultad de Ingeniería Mecánica y Eléctrica (F.I.M.E) por el apoyo brindado durante mis estudios de maestría. Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado mediante una beca de estudios de tiempo completo.

Quiero agradecer enormemente al Dr. José Arturo Berrones Santos, director de mi tesis, por la oportunidad de pertenecer a su grupo de investigación, por haberme guiado durante todo este tiempo, por aconsejarme y siempre estar al pendiente de nosotros, sus tesistas. Gracias por su amistad.

Agradezco a todos los profesores y compañeros del PISIS por ser partícipe de esta gran aventura. En especial al Dr. Igor Litivinchev (PISIS) y al Dr. Francisco Almaguer Martínez de la Facultad de Ciencias Físico Matemáticas, por ser parte de mi comité de tesis y por guiarme en la elaboración de la misma Así mismo, agradezco a Jonás por su paciencia y apoyo. A mis compañeros y amigos de generación 2012-2014, gracias.

Gracias dios por tantas bendiciones.

Resumen

Los problemas de optimización combinatoria aparecen en un gran número de aplicaciones tanto en la vida real como en casos de estudio académicos. Problemas que en su mayoría son difíciles de resolver conforme crece el tamaño de los casos, por lo que, no exite metodología que los resuelva, o bien, existe pero el tiempo de respuesta es muy elevado en casos a gran escala. Uno de estos problemas difíciles y retadores en el área académica es el problema de la mochila cuadrática, el cual es elegido para realizar nuestras pruebas. Es de suma importancia desarrollar y explorar nuevas estrategias que ayuden a disminuir el espacio de búsqueda y por ende los tiempos de respuesta. El método propuesto esta basado en la teoría de campo medio, la cual ha sido poco trabajada en problemas de optimización combinatoria, más sin embargo, en la literatura los pocos trabajos reportados muestran resultados muy prometedores. En el presente trabajo se hace una comparativa entre un híbrido de campo medio más una búsqueda local y un híbrido de un constructivo muy conocido, voraz o greedy (en inglés) aunado a una búsqueda local.

Juan Antonio Banda Moreno.

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

Universidad Autónoma de Nuevo León.
Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

Estudio de un algoritmo heurístico híbrido basado en la teoría de campo medio aunado a una búsqueda local para el problema de la mochila cuadrática

Número de páginas: 45.

Objetivos y método de estudio: El objetivo principal de esta tesis consiste en evaluar el desempeño de la teoría de campo medio más una búsqueda local a través de una comparativa entre este híbrido y el híbrido de voraz más una búsqueda local. Para lograr nuestro objetivo, es elegido el problema de la mochila cuadrática en casos de gran tamaño que no han sido reportados en la literatura hasta la fecha.

Contribuciones y conlusiones: El problema de la mochila cuadrática es un problema de optimización combinatoria discreta no lineal difícil de resolver, cuando el tamaño de los casos de estudio son grandes. En el presente trabajo se muestra un buen desempeño de la estrategia propuesta híbrida a comparación del híbrido de voraz. Los resultados obtenidos en cuanto a calidad de la solución son muy similare he inclusive algunos son mejores; en cuanto a tiempo en ciertos casos campo medio favorece a la reducción de tiempos de respuesta.

Firma del asesor: _____

Dr. José Arturo Berrones Santos

Capítulo 1

Introducción

Una gran cantidad de problemas reales pueden ser representados como modelos de optimización, los cuales consisten en la búsqueda de una “mejor“ configuración de un conjunto de variables de tal manera que se logre alcanzar algún objetivo o meta [7], a esta ”mejor“ configuración le llamaremos solución óptima. Este tipo de problemas se clasifican en dos categorías: aquellos donde las soluciones están codificadas con variables de tipo valor real y aquellos en los que se codifican soluciones con variables de tipo discretas [7]. Entre estos últimos, encontramos una clase de problemas denominados problemas de optimización combinatoria (COP, por sus siglas en inglés combinatorial optimization problems). En un COP lo que se quiere encontrar es un objeto de un conjunto finito [28], de tal forma que ciertas condiciones se satisfacen [35]. Este objeto es típicamenteun un número entero, un subconjunto, una permutación o una estructura gráfica [7]. En el presente trabajo se estará hablando muy frecuentemente de un objeto como vector solución, que consta de variables binarias, es decir, ceros y unos. Donde los ceros representan aquellos valores que están inactivos y los unos representan aquellos valores que están activos, por lo que el vector solución tiene la siguiente forma:

1	1	0	0	...	1	0	1	0

Muchos de los problemas de optimización combinatoria tienen objetos en los que habitualmente el número de sus elementos es muy elevado, haciendo impracticable la evaluación de todas sus soluciones para determinar el valor óptimo [20]. A este

tipo de proplemas se les llama NP-duros, los cuales son difíciles de resolver cuando el tamaño de los casos es significativamente grande. En complejidad computacional, un problema es llamado NP-duro si y solo si, éste es al menos tan díficil como cualquier problema NP, es decir, cada problema NP puede ser reducido a éste. Un problema de decisión es NP si y solo si, cualquier solución dada del problema puede ser verificada en un tiempo polinomial [23].

Los problemas de optimización combinatoria más conocidos en la literatura son el problema del agente viajero [4, 14], mochila cuadrática [16, 32, 38], ruteo de vehículos [37, 17], secuenciación de tareas [31], p-mediana [33], entre otros. Este tipo de problemas tienen aplicaciones en diversos campos como la economía, el comercio, la ingeniería, la industria, la medicina, etc. Los problemas de optimización pueden ser resueltos por un método exacto o un método heurístico. Inclusive, existen metodologías que combinan ambas estrategias, aprovechando lo mejor de cada una de ellas. Este tipo de combinaciones se realizan con el fin de disminuir el espacio de búsqueda y/o los tiempos de respuesta [36, 35], del problema en cuestión. Los métodos exactos, a diferencia de los métodos heurísticos, obtienen la solución óptima del problema que a menudo es computacionalmente muy costoso, es decir, generalmente los métodos exactos tardan mucho tiempo en resolver el problema; esto debido a que la mayoría de los casos reales y de estudio son de gran tamaño. Algunas de las estrategias exactas más populares y usadas para resolver problemas combinatorios son el método ramificación y cota [18], y el método de ramificación y corte [21]. Los métodos heurísticos generan soluciones de buena calidad en un tiempo razonable de cómputo, más sin embargo, no garantizan encontrar el valor óptimo [36]. Se ha observado en la literatura que la implementación de una buena estrategia heurística, casi siempre brindan soluciones satisfactorias en sucesos en los que un método exacto no puede dar respuesta al caso de estudio. Cuando un problema es de naturaleza tal que no se conoce ningún método exacto que permita encontrar su solución óptima, o aunque exista, el uso de éste es computacionalmente muy costoso, se recurre a un método heurístico. Entre los algoritmos heurísticos básicos solemos distinguir los métodos constructivos (MC) y métodos de búsqueda local (BL) [20, 8], estrategias

que constituyen la base de los métodos metaheurísticos. Un metaheurístico es una estrategia que guía a otro heurístico para escapar de óptimos locales e incrementar su desempeño [36, 13], con la esperanza de encontrar el valor óptimo global (solución óptima).

Los métodos constructivos generan soluciones desde cero, mediante la adición de componentes hasta que una solución es completada, frecuentemente son usadas como una solución inicial dentro de otra heurística que mejora la calidad de la solución [20]. Los MC, también, son usados como soluciones iniciales en algunos algoritmos exactos para disminuir el espacio de búsqueda y los tiempos de respuesta. Uno de los métodos constructivos más conocido y usado en problemas combinatorios es la heurística constructiva voraz (V), llamado en la lengua inglesa como greedy. Los métodos de búsqueda local parten de alguna solución inicial, e iterativamente intentan reemplazar la solución actual por una mejor solución, en un vecindario definido apropiadamenta a partir de la solución actual. El método de búsqueda local finaliza cuando, para una solución no existe ninguna solución accesible que la mejore [1, 20]. Hablaremos un poco más detallado en la secciones 2.1 y 2.2 sobre el algoritmo voraz y el algoritmo de búsqueda local simple, respectivamente.

Los métodos heurísticos y metaheurísticos más usados para resolver problemas combinatorios son: búsqueda local simple o básica, búsqueda tabú, recocido simulado, GRASP (por sus siglas en inglés, greedy randomized adaptive search procedure), algoritmos genéticos, entre otros; para una descripción detallada sobre estos métodos ver [8]. Todos los métodos mencionados anteriormente dependen de una distribución de probabilidad arbitraria, que usualmente tiene poca o ninguna relación con el problema. Por ejemplo, en los algoritmos genéticos la variabilidad de un vector solución, denominado individuo, viene dada por un operador llamado mutación. La mutación hace ligeros cambios en los individuos y es generada por medio de una distribución de probabilidad uniforme, la cual no tiene nada que ver con el problema.

En estos últimos años, se han propuesto una serie de algoritmos evolutivos que guían la exploración del espacio de búsqueda, mediante la construcción de mode-

los probabilísticos encontrando soluciones prometedoras. Estos algoritmos han demostrado funcionar muy bien en una amplia variedad de problemas de optimización [29, 30]. En este contexto de los modelos probabilísticos, se introduce la teoría de campo medio (CM) para generar soluciones de buena calidad. La idea principal de esta teoría es sustituir una distribución difícil por otra más sencilla [26]. Nos referimos a una distribución difícil, a aquella distribución que es desconocida o su estructura no es fácil de tratar, como sucede en la mayoría de los problemas de optimización combinatoria.

1.1 Descripción del Problema

En trabajos previos se mostró que campo medio ofrece buenas soluciones en el problema de la mochila cuadrática en casos de tamaños superiores a los reportados en la literatura. Además, en el artículo [25] se muestra un gran desempeño de la teoría de campo medio en combinación con programación lineal, para el problema de la mochila con N variables y M restricciones. También, se observó un buen rendimiento de campo medio en el problema de cubrimiento de conjuntos [24]. La teoría de campo medio ha sido poco explorada hasta la fecha en problemas combinatorios, por lo que en esta tesis se muestra el desempeño de campo medio incorporado en una búsqueda local simple. El desempeño de este híbrido se evalua poniendolo en competencia en contra de un híbrido voraz más una búsqueda local simple. El algoritmo voraz fue seleccionado para realizar las comparaciones puesto que en la literatura se ha mostrado que el método voraz ofrece muy buenos resultados en el problema de la mochila.

1.2 Motivación y Justificación

Hoy en día, diversas áreas del conocimiento e industria exigen respuestas casi inmediatas para sus problemas de optimización. Gran parte de estos problemas

tienen estructuras de modelos de optimización combinatoria, de los cuales en su mayoría no existe alguna metodología que aproxime su solución, o bien, la respuesta de un método exacto es muy tardada. Por esta razón es de suma importancia, desarrollar algoritmos que resuelvan eficientemente el problema [35] de tal manera que se disminuya el espacio de búsqueda y que las soluciones reportadas sean de muy buena calidad en un tiempo razonable de cómputo.

Capítulo 2

Marco Teórico

Un problema de optimización combinatoria (COP, por sus siglas en inglés combinatorial optimization problem), es aquel en el que la solución consiste de una combinación de componentes únicas, seleccionada típicamente de un conjunto finito [19]. El objetivo es encontrar la combinación perfecta de componentes que brinden el mayor beneficio o reduzcan lo más que se pueda los costos, es decir, se quiere encontar una solución óptima combinatoria que maximice o minimice una función objetivo, sujeta a ciertas restricciones.

Dos ejemplos clásicos de optimización combinatoria son el problema de la mochila simple (KP, por sus siglas en inglés knapsack problem) [34] y el problema del agente viajero (TSP, por sus siglas en inglés traveling salesman problem) [14]. El primer problema consiste de n artículos con un peso y un beneficio asociado a cada uno de éstos. La mochila tiene cierta capacidad, por lo que el objetivo es seleccionar aquellos artículos que proporcionen el mayor beneficio sin exceder su peso máximo. Las soluciones que sobrepasen la capacidad de la mochila se dice que son infactibles, es decir, las soluciones son ilegales o inválidas. En el segundo problema se tienen un conjunto de ciudades con algún número de rutas (o planes de vuelo) entre varios pares de ciudades. Cada ruta tiene un costo asociado. El agente debe realizar un recorrido comenzando en una ciudad A, visitando todas las ciudades al menos una vez y finalmente regresar a A. El objetivo es encontrar el recorrido de menor costo de tal manera que se visiten cada una de las ciudades por lo menos una vez, regresando a la ciudad de origen.

Como ya se mencionó anteriormente, el objetivo de los problemas de optimización combinatoria es encontrar el máximo o mínimo, según sea el caso, de una determinada función sobre un conjunto finito de soluciones, cuyas variables de decisión son discretas [20] sujeto a determinadas condiciones. Habitualmente en este tipo de problemas, el número de elementos del conjunto es muy elevado, haciendo impracticable la evaluación de todas sus soluciones debido a que el tiempo de respuesta es muy alto. Por esta razón, se han desarrollado y se siguen desarrollando estrategias que generan soluciones de buena calidad en poco tiempo, no necesariamente la óptima.

Las técnicas usadas para resolver los problemas de optimización combinatoria pueden ser clasificadas en dos categorias: algoritmos exactos y algoritmos heurísticos. Los algoritmos exactos garantizan encontrar la solución óptima, para cualquier caso de este tipo de problemas, de entre un conjunto finito de soluciones [22, 28]. Sin embargo, para los problemas de optimización combinatoria que son NP-difícil no existe un algoritmo que los resuelva en tiempo polinomial, por lo que el método exacto puede necesitar un tiempo de cálculo exponencial en el peor de los casos. Ésto a menudo conduce a tiempos de cálculo demasiado altos para propósitos prácticos. Por ello, en los últimos 30 años, los algoritmos heurísticos cada vez más han sido altamente aceptados para resolver este tipo de problemas. La palabra Heurística se deriva del verbo griego heuriskein que significa “encontrar”. Los heurísticos obtienen buenas soluciones en un tiempo aceptable, no necesariamente la óptima, es decir, los métodos heurísticos sacrifican la garantía de encontrar soluciones óptimas con el fin de obtener buenas soluciones en un tiempo significativamente menor al de los procedimientos exactos. Los enfoques exactos que han tenido mucho éxito para resolver los problemas de optimización combinatoria son: el método de ramificación y acotamiento (en inglés, branch and bound), programación dinámica (dynamic programming, en inglés), programación restrictiva (constraint programming, en inglés) y programación entera lineal (integer linear programming, en inglés). Los métodos heurísticos y metaheurísticos se basan generalmente en dos principios básicos: una heurística constructiva y un método de búsqueda local. Un metaheurístico se de-

fine formalmente como un proceso de generación iterativa que guía una heurística subordinada, combinando de forma inteligente distintos conceptos para explorar y explotar el espacio de búsqueda, algunas estrategias de aprendizaje son usadas con el fin de encontrar soluciones cercanas al valor óptimo de manera eficiente [27]. Los metaheurísticos fueron creados para evitar estancarse en óptimos locales, explorando regiones que aún no han sido visitadas con la esperanza de encotrar al óptimo global. Algunas de estas estrategias metaheurísticas más conocidas para resolver problemas de optimización combinatoria reportados en la literatura son: optimización basada en colonia de hormigas, computación evolutiva, algoritmos genéticos, recocido simulado, búsqueda tabú, GRASP (por sus siglas en inglés, Greedy Randomized Adaptive Search Procedure), entre otras [8]. Los heurísticos constructivos generan soluciones desde cero mediante la adición de componentes a una solución inicial vacía mediante una estrategia bien definida. Este proceso se realiza hasta que una solución es completada o un criterio de paro se satisface. Una clase muy conocida de los heurísticos constructivos es el heurístico voraz (greedy), el cual se explicará con más detalle en la sección 2.1 . Las heurísticas constructivas son a menudo rápidas, sin embargo, a menudo regresan soluciones de calidad inferior en comparación con los algoritmos de búsqueda local. Los algoritmos de búsqueda local parten de una solución inicial y de forma iterativa tratan de sustituir la solución actual por una solución mejor en un vecindario de la solución actual definido apropiadamente, en la sección 2.2 se hablará un poco más sobre esta metodología.

En los útimos años, se han reportado un gran número de algoritmos que no siguen la estructura pura del método metaheurístico o heurístico en sí, si no que combinan varias componentes de diversos algoritmos. Estos enfoques son comúnmente llamados híbridos metaheurísticos o híbridos heurísticos, los cuales puede proporcionar un mejor comportamiento y una mayor flexibilidad cuando se trata de problemas del mundo real de gran tamaño. La efectividad de estos algoritmos puede lograrse por ejemplo, combinando estrategias complementarias de las metaheurísticas por un lado y por el otro, combinando ciertos elementos de una metaheurísticas con alguno de los métodos exactos como el de ramificación y cota o progra-

mación cuadrática. La principal motivación detrás de la hibridación de diferentes metodologías es, beneficiarse de la sinergia de ciertas cualidades de los métodos en base a los requerimientos del problema a tratar. Hay tipos de hibridaciones que han demostrado ser exitosas para muchas aplicaciones difíciles. Sin embargo, un híbrido no puede funcionar bien para cualquier tipo de problema de optimización, es decir, no se puede generalizar el uso de un determinado híbrido para todo caso de estudio [6], como sucede en cualquier heurístico.

2.1 Método voraz (greedy)

Muchos autores se refieren a los algoritmos greedy como algoritmos voraces, algoritmos devoradores, algoritmos glotones, algoritmos miopes, etc [12]; nosotros en el presente trabajo lo denominaremos voraz (V). El algoritmo voraz es uno de los enfoques más simples y conocidos para resolver problemas de optimización (minimizar o maximizar) en el que se quiere determinar el óptimo global de una función dada en una serie de pasos, en cada etapa del algoritmo podemos elegir entre un conjunto de posibles decisiones. En este método la elección de las decisiones óptimas se hace sobre la información que se tiene a la mano, sin preocuparse de los efectos que estas decisiones puedan causar en el futuro. Estos algoritmos son fácil de implementar, más aún, los tiempos de respuesta son bastante razonables [15]. Esta estrategia ha sido empleada en una gran diversidad de problemas, obteniéndose en algunos casos especiales la solución óptima, en otros casos simplemente buenas soluciones y, en los más, soluciones mediocres [12]. Un ejemplo de una heurística voraz es el vecino más cercano, para el famoso problema del agente viajero [15]. En años recientes, muchos algoritmos modernos o heurísticos han sido propuestos en la literatura así como mejoras del algoritmo voraz. De hecho, muchos de los metaheurísticos como el recocido simulado y algoritmos genéticos estan basados sobre la estrategia voraz [3].

A continuación, se muestra el algoritmo de voraz básico tomado de [12], donde Candidatos es la lista de todos los objetos ordenados en base a la función de evalu-

ación, definida apropiadamente:

Procedimiento VORAZ

Solución=ø;

Definir Candidatos;

MIENTRAS (solución no completa Y candidatos no vacío) HACER

Seleccionar elemento de candidatos que optimice localmente

SI (Elemento no viola restricciones) ENTONCES

Solución=Solución+elemento;

Candidatos=Candidatos-elemento;

END; {Si}

END; {Mientras}

SALIDA (Solución);

Dos características importantes que hacen muy popular al método voraz son la simple implementación y eficiencia, algunas veces puede lograr soluciones óptimas para algunos problemas de optimización. Por ejemplo, en el probema de selección de actividades, problema de la mochila y en el problema del árbol de expansión mínima, el algoritmo básico voraz puede obtener la solución óptima realizando una serie de elecciones voraces bien definidas. Sin embargo, en algunos problemas de optimización, especialmente los de optimización combinatoria NP-difícil, las solución del algoritmo voraz simple esta muy alejada de una solución satisfactoria por lo que

novedosas estrategias de mejoramiento de voraz han surgido recientemente, para mayor información sobre éstas ver [3].

El método voraz, como ya mencionamos en el primer capítulo 1, es una heurística constructiva la cual genera soluciones desde cero, mediante la adición de componentes hasta que una solución es completada [20]. El voraz selecciona en cada iteración el candidato con mayor valor de una función de evaluación, normalmente es muy rápida y a menudo devuelve soluciones razonablemente buenas [16, 10]. La función de evaluación definida para nuestro caso de estudio, conocido en la literatura como el problema de la mochila cuadrática es: la suma de todos los valores asociados con el artículo i dividido entre su respectivo peso, para mayor información ver [16]. De manera que, para generar una solución inicial de voraz en el problema de la mochila cuadrática se tienen que seguir los siguientes pasos:

1. Ordenamos los valores de la función de evaluación de manera descendente.

2. Iterativamente incorporamos artículos a la mochila en base a la función de evaluación, siempre y cuando no exceda la capacidad máxima. Si al incorporar un artículo a la mochila, éste hace que la suma de todos los pesos agregados hasta el momento sea mayor que el permitido, se retira y se verifica con el siguiente mejor de la función de evaluación. El proceso se repite hasta probar con todos los artículos.

2.2 Búsqueda local

El método de búsqueda local más básico es usualmente llamado búsqueda local de mejora iterativa, ya que cada movimiento sólo se realiza si la solución resultante es mejor que la actual. El algoritmo se detiene cuando se ha alcanzado el mínimo local (o máximo local, según sea el caso) o algún criterio de paro se satisfaga [9]. Las técnicas de búsqueda local son heurísticas iterativas que han sido muy populares como medio para encontrar soluciones en problemas de optimización combinatoria

difíciles. Inumerables ejemplos de problemas combinatorios podrían citarse en los que estos métodos han sido ampliamente sugeridos como apropiados, o para los cuales estas estrategias puede ser usadas para mejorar una solución generada por algún otro método heurístico. Típicamente, ésta heurística comienza con una solución factible e iterativamente mejora la calidad de la solución. En cada paso, considera operaciones locales para mejorar la solución. Este método trabaja con la hipótesis de que existe un vecindario definido sobre el espacio de todas las posibles soluciones [2].

Los procedimientos de búsqueda local, también llamados de mejora, se basan en explorar el entorno o vecindad de una solución. Utilizan una operación básica llamada movimiento que, aplicada sobre los diferentes elementos de una solución, proporciona las soluciones de su entorno. La definición de entorno/movimiento, depende en gran medida de la estructura del problema a resolver, así como de la función objetivo [20]. El movimiento definido para nuestro caso de estudio, el problema de la mochila cuadrática, es meter un elemento que no está en la mochila por otro que si está dentro.

Los dos métodos básicos comúnmente usados para seleccionar un mejor vecindario son el de mejor mejora y el de primer mejora, a continuación se describen cada uno de ellos, para mayor información consultar [2, 20].

1. **Mejor mejora:** Es un método de "descenso rápido", explora exhaustivamente el vecindario y devuelve la solución que es mejor que la solución mejor actual.

2. **Primer mejora:** Esta estrategia explora el entorno vecindario y devuelve la primera solución que es mejor que la solución mejor actual.

Ambos métodos se detienen en óptimos locales y en ocasiones ese valor es el óptimo global. Por lo tanto, su fuerte rendimiento depende de la definición de la estructura de vecindad y de la solución inicial [9]. La estrategia usada en el presente trabajo es el de primer mejora aunada a un método voraz y aunada a una heurística basada en campo medio, que más adelante se explicará en que consiten ambas hibridaciones.

El rendimiento de los procedimientos de búsqueda local de mejora iterativa

simple es en general insatisfactorio, debido a que la calidad del mínimo local obtenida depende en gran medida del punto de partida en el proceso de búsqueda local. Como la cuenca de atracción de un mínimo global es generalmente desconocido, la búsqueda local de mejora iterativa simple puede terminar en un mínimo local de mala calidad. Una primera estrategia sencilla de extender la búsqueda local de mejora iterativa consiste en iterativamente comenzar en diferentes puntos de arranque, los cuales pueden ser generados aleatoriamente. Para este tipo de búsqueda local multiarranque es posible obtener mejores resultados. Sin embargo, por lo general aún esta estrategia está lejos de ser satisfactoria para algunos problemas de optimización. Por lo tanto, varias técnicas metaheurísticas se han desarrollado con el objetivo de agregar una componente de exploración para la búsqueda local de mejora iterativa. Esta componente de exploración es responsable de guiar la exploración del espacio de búsqueda en la indagación de mejores y mejores mínimos locales[9]. Algunas de las más importantes metaheurísticas en las que se ha usado estrategias de búsqueda local son: recocido simulado, búsqueda tabú, GRASP (por sus siglas en inglés, greedy randomized adaptive search procedure), entre otras.

A continuación se muestra el algoritmo básico de una búsqueda local, el cual parte de una solución incial $\vec{x}_0$ generada por un algoritmo heurístico constructivo y que es mejorada por la búsqueda local, regresando como salida el vector solución $\vec{x}$ quién será el óptimo local encontrado.

Procedimiento BL

$\vec{x} = \vec{x}_0$;

MIENTRAS exista $\vec{y} \in N(\vec{x})$ tal que $f(\vec{y}) > f(\vec{x})$, ENTONCES $\vec{x} = \vec{y}$.

SALIDA $(\vec{x})$

2.3 Inferencia variacional

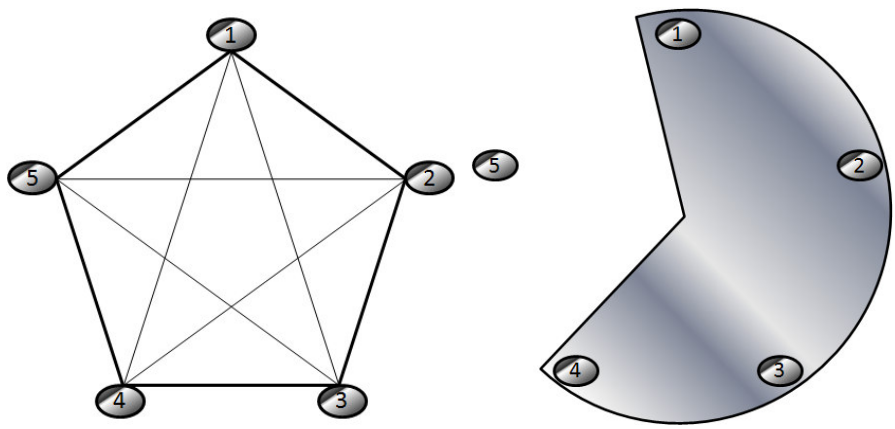
Los métodos variacionales tienen sus origenes en el siglo XVIII con los trabajos de Euler, Lagrange y otros sobre cálculo de variaciones. El cálculo variacional es una rama de las matemáticas que consiste en buscar máximos o mínimos de funcionales continuos definidos sobre algún espacio funcional. Estas estrategias tienen una amplia aplicabilidad e incluyen áreas tales como los métodos de elementos finitos (Kapur, 1989) y la entropía máxima (Schwarz, 1988). Los métodos variacionales encuentran soluciones aproximadas mediante la restricción del rango de funciones sobre la que se realiza la optimización. Teoría y algunos conceptos que se tratan en esta sección, fueron tomados de los libros [5] y [26], los cuales consideramos son los más aceptados en la literatura para poder explicar la idea central de la teoría de campo medio y que además, son un poco más sencillos de comprender.

2.3.1 Campo medio

Una de las más simples y prominentes aproximaciones de una distribución difícil de tratar, está basados sobre el llamado método de campo medio, el cual tiene una larga historia en física estadística. La idea principal de la teoría de campo medio es centrarse en una determinada entidad, y sustituir todas las interacciones con las otras entidades por una interacción media o eficaz. Una representación gráfica de este concepto lo podemos ver e interpretar con mayor claridad en la imagen 2.1, donde se muestra la forma clásica de visualizar un problema de optimización mediante un grafo. El grafo que se ilustra es completo, es decir,un grafo simple donde cada par de vértices está conectado por una arista.

2.3.2 Campo medio variacional

Este método aproxima una distribución $P(\vec{x})$ difícil de tratar mediante una distribución $Q(\vec{x})$, la cual viene de una familia M de distribuciones más sencillas



(a) Problema de optimización.

(b) Idea principal de la toría de campo medio.

Figura 2.1: Representación gráfica de un problema de optimización mediante un grafo.

y que además, debe ser lo suficientemente rica y flexible que pueda proporcionar resultados muy parecidos a los de la verdadera distribución, donde $\vec{x} = (x_1, x_2, ..., x_N)$ es un vector de variables aleatorias. La distribución Q es elegida de tal forma que se minimiza una cierta medida de distancia $D_{KL}(Q, P)$, es decir, se selecciona una distribución más sencilla que se parezca lo más posible a P. Dicha distancia es conocida en la literatura como la distancia de Kullback Leibler, de la cual no se profundizará en el presente trabajo.

Una distribución de probabilidad puede estar asociada con la siguiente transformación:

$$P(\vec{x}) = \frac{1}{Z} exp(\frac{-V}{kT})$$

donde Z es un factor de normalización y kT es una constante. P es conocida como la distribución de Boltzmann, la cual proporciona la máxima entropía del sistema. Entre más grande sea el valor de la entropía, el sistema se encontrará en desorden como consecuencia podemos obtener mayor información del mismo. Por ello, usaremos esta distribución que contendrá la función objetivo y sus restricciones.

La distribución P es en general difícil de tratar por su complicada estructura,

por lo que se propone un modelo probabilístico de campo medio Q, para las variables de decisión, más sencillo que P. Asumiendo que cada x_i son variables independientes, entonces Q puede expresarse como el producto de sus distribuciones marginales:

$$Q(\vec{x}) = \prod_{i=1}^{N} q_i(x_i)$$

donde las x_i son variables de decisión binarias. Esta forma factorizada de la inferencia variacional corresponde a un trabajo de aproximación desarrollado en física llamado teoría de campo medio (Parisi, 1988).

De manera general, consideremos el siguiente problema de optimización:

$$min \quad f(\vec{x}), \qquad s.a \qquad g_k(\vec{x}) \leq 0, \quad h_l(\vec{x}) = 0 \tag{2.1}$$

en donde $\vec{x}$ es un vector de variables de decisión binaria, g_k y h_l son restricciones de desigualdad e igualdad, respectivamente. La forma más general para las marginales q_i de este problema de optimización es:

$$q_i(x_i, m_i) = 1 + (2m_i - 1)x_i - m_i$$

donde las m son parámetros del campo medio continuo, $m \in [0, 1]$. Cada m_i es identificada como los valores esperados de x_i bajo Q, es decir, $m_i = \langle x_i \rangle_Q$; y pueden ser selecionadas por la minimización de la divergencia Kullback-Leibler entre la distribución Q y P. $D_{KL}(Q||P) = \langle \ln Q \rangle - \langle \ln P \rangle$, donde $\langle\rangle$ representan los promedios con respecto a la distribución tratable Q.

Introduciendo la entropía $S_Q = -kT\langle \ln Q \rangle$, se obtiene el problema variacional $min\ F_Q$, donde

$$F_Q = \frac{1}{kT}[\langle V \rangle - S_Q]$$

es la energía libre variacional de la distribución Q. Sin pérdida de generealidad, la constante kT se define como $kT = 1$.

La relajación continua del problema (2.1) es equivalente a sus promedios bajo la distribución de campo medio Q, es decir:

$$min \quad f(\vec{m}), \qquad s.a \qquad g_k(\vec{m}) \leq 0, \quad h_l(\vec{m}) = 0 \tag{2.2}$$

Una expresión para $\langle V \rangle$ puede ser construida en términos de la relajación Lagrangiana como se muestra a continuación:

$$\langle V \rangle = \mathcal{L} = f(\vec{m}) + \sum_l \lambda_l h_l(\vec{m}) + \sum_k \mu_k g_k(\vec{m}) \tag{2.3}$$

donde λ_l y $\mu_k \geq 0$, son los multiplicadores de Lagrange. En la sección 2.4 se da una breve explicación sobre estos multiplicadores, también conocidos como multiplicadores de las condiciones de Karush-Kuhn-Tucker.

Usando la independencia de las m_i con respecto a Q, la entropía variacional $S_Q = -\sum q_i \ln q_i$ está dada como:

$$S_Q = -\sum_i [(1 - m_i)\ln(1 - m_i) + m_i \ln m_i],$$

entonces el problema variacional para las m's se escribe como:

$$min\{f(\vec{m}) + \sum_l \lambda_l h_l(\vec{m}) + \sum_k \mu_k g_k(\vec{m}) + \sum_i [(1 - m_i)\ln(1 - m_i) + m_i \ln m_i]\}$$

Debido a que las variables m_i son independientes bajo campo medio, V se define como la expresión 2.3 para cualquier problema 2.1, siempre y cuando se cumplan las condiciones establecidas. El problema variacional se reduce a un conjunto de ecuaciones independientes para cada una de las $m's$.

$$mi = \frac{1}{1 + exp[\partial_i \mathcal{L}_{\vec{\lambda},\vec{\mu}}(\vec{m})]} \tag{2.4}$$

2.4 Condiciones de Karush-Kuhn-Tucker

Los problemas de programación lineal, en los que las restricciones y la función a optimizar son lineales. Aunque los problemas de programación lineal son muy comunes y cubren un amplio rango de aplicaciones, en la vida real uno se tiene que

enfrentar con cierta frecuencia a otro tipo de problemas que no son lineales. Cuando el conjunto de restricciones, la función objetivo, o ambos, son no lineales, se dice que se trata de un problema de programación no lineal (PPNL)

Estas condiciones deben ser satisfechas por la solución óptima de cualquier problema lineal y la mayoría de los problemas no lineales. Constituyen la base para el desarrollo de muchos algoritmos computacionales y proporciona un criterio de parada para muchos otros, permitiendo establecer cuando ha sido alcanzado un óptimo local restringido. Las condiciones de Karush-Kuhn-Tucker generalizan la condición necesaria desarrollada para problemas no restringidos a los problemas con restricciones.

Definición. Condiciones de Karush-Kuhn-Tucker (CKKT) . EL vector $\vec{x} \in \mathbb{R}^n$ satisface las CKKT para PPNL si existe un par de vectores $\lambda \in \mathbb{R}^l$ y $\mu \in \mathbb{R}^m$ tales que

$$\nabla f(\vec{x}) + \sum_{k=1}^{l} \lambda_k \nabla h_k(\vec{x}) + \sum_{j=1}^{m} \mu_j \nabla g_j(\vec{x}) = 0 \tag{2.5}$$

$$h_k(\vec{x}) = 0, k = 1, ..., l \tag{2.6}$$

$$g_j(\vec{x}) \leq 0, j = 1, ..., m \tag{2.7}$$

$$\mu_j g_j(\vec{x}) = 0, j = 1, ..., m \tag{2.8}$$

$$\mu_j \geq 0, j = 1, ..., m \tag{2.9}$$

Los vectores μ y λ se denominan los multiplicadores de Kuhn-Tucker. La condición 2.8 es conocida con el nombre condición de complementariedad, la condición 2.9 requiere la no negatividad de los multiplicadores, y es llamada condición de factibilidad dual, y 2.6-2.7 se denominan condiciones de factibilidad primal [11].

Capítulo 3

Caso de Estudio

El caso de estudio que se aborda en el presente trabajo para validar el desempeño de campo medio, pertenece al conjuto de problemas de optimización combinatoria en los cuales el tiempo de respuesta incrementa conforme aumenta el tamaño de los casos. Este modelo es conocido en la literatura como el problema de la mochila cuadrática (QKP por sus siglas en inglés, quadratic knapsack problem), el cual fue introducido por Gallo [39, 32]. El problema de la mochila cuadrática es usada en múltiples aplicaciones en las diversas áreas del conocimiento como en problemas de teoría de grafos (máximo clique, problema del b-clique máximo ponderado), problemas de localización de instalaciones, separación de desigualdades válidas para el problema de ruteo de vehículos, entre otros.

Debido a su dificultad para resolver cuando el tamaño de los casos es grande, su simple estructura y su extensa aplicabilidad en casos reales, el QKP es ampliamente estudiado en investigación de operaciones. Por tales motivos, han surgido diferentes técnicas o enfoques que tratan de resolverlos en determinados tamaños de los casos. La mayoría de los métodos exactos que han surgido están diseñados en base a los enfoques del método de ramificación y poda, agregando estrategias que permitan reducir el espacio de búsqueda lo más posible. Además, numerosas cotas superiores han sido obtenidas mediante derivados de planos superiores, linealización, reformulaciones, relajaciones Lagrangianas, descomposición Lagrangiana, relajación semidefinida, estrategias de reducción, etc [38]; cotas que permiten reducir la búsqueda o conociendo una cota inferior, podemos saber en que intervalo se encuentra la

solución óptima. Vale la pena mencionar que el algoritmo desarrollado por Pisinger, basado sobre la relajación/descomposición Lagrangiana y reducción agresiva, es uno de los algoritmos exactos más eficaces para resolver el QKP hasta la fecha, el cual ha resuelto algunos casos de un máximo de 1500 variables binarias en un tiempo considerable [39].

También se han propuesto un sin fin de heurísticos que brindan buenas soluciones en un plazo razonable. El heurístico más reciente propuesto por Xie y Liu, conocido como el algoritmo de Mini-Swarm, computacionalmente ha demostrado ser muy efectivo y capaz de resolver a optimalidad un alto porcentaje (94.9 %) de instancias de prueba de Billionnet y Soutif con un máximo de 200 variables en un tiempo muy razonable de cómputo [39].

3.1 Problema de la Mochila Cuadrática

El QKP puede ser definido como sigue: asumamos que tenemos n artículos, donde cada artículo j tiene un peso entero positivo w_j. Además, se tiene una matriz con valores enteros no negativos $P = \{p_{ij}\}$; donde p_{jj} es el beneficio obtenido si el artículo j es seleccionado, y $p_{ij} + p_{ji}$ es el beneficio obtenido si los artículos i y j son ambos seleccionados para $i < j$. Este modelo es una generalización del problema de la mochila (KP por sus siglas en inglés, knapsack problem), que surge cuando $p_{ij} = 0, \forall i \neq j$. El problema consiste en seleccionar de un subconjunto $S \subseteq \{1, ..., N\}$ de elementos potenciales, aquellos elementos que maximicen el beneficio global, sin exceder el peso máximo de la mochila [39, 32, 38]. El problema puede ser formulado matemáticamente como sigue:

$$maximizar \qquad \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j \tag{3.1}$$

$$sujeto \quad a: \qquad \sum_{j \in N} w_j x_j \leq c, \tag{3.2}$$

$$x_j \in \{0, 1\}, \qquad j \in N. \tag{3.3}$$

La matriz de beneficios es simétrica, es decir, $p_{ij} = p_{ji}$.

3.1.1 Aplicaciones

A continuación mostraremos una de las aplicaciones que pudiese ser la más sencilla de explicar e interpretar. Igualmente, es muy útil en ciudades altamente pobladas en las que la congestión vehicular es recurrente. Por ello, es necesario la implementación de alternativas de transporte que reduzcan los tiempos de traslado, el uso de vehículos que contaminan el medio ambiente, costos, entre otros aspectos. Una de las opciones de transporte más económicas y rápidas es el uso del metro. Para la construcción de una línea del metro, los gobernantes deben seleccionar de un subconjunto de estaciones posibles, aquellas que reúnan un mayor número de usuarios y que además, la ruta le sea de utilidad, para llegar en el menor tiempo posible. Asimismo, la construcción de las estaciones no debe exceder el presupuesto estimado del gobierno.

Como podemos observar en la figura 3.1

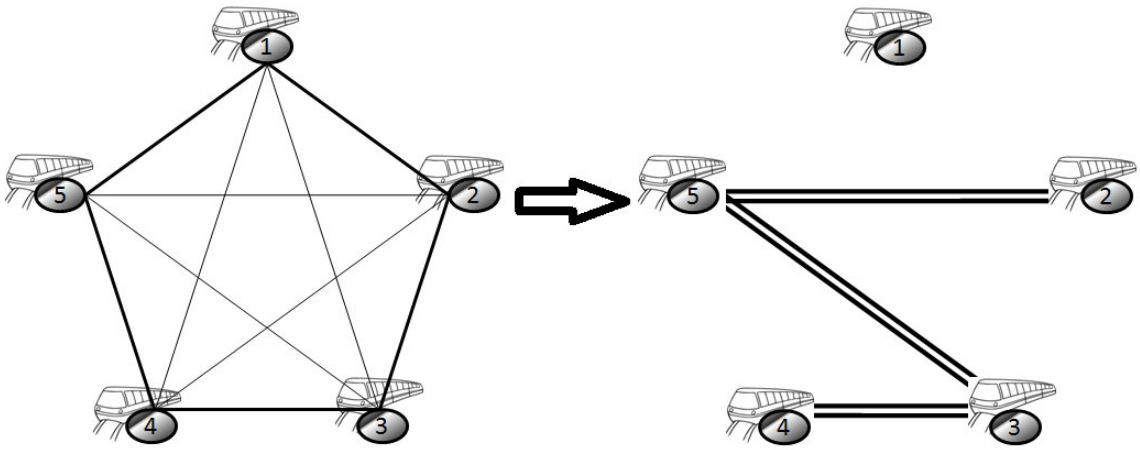


Figura 3.1: Grafo.

Capítulo 4

Métodos de Solución

La metodología propuesta para validar el desempeño de la teoría de campo medio en el problema de la mochila cuadrática con otras heurísticas hibridarlo con una búsqueda local simple, como no existe un método exacto que brinde resultados para tamaños grandes en un tiempo razonable, por lo que se compara con un híbrido voraz y una búsqueda local. A continuación se muestran los algoritmos que se desarrollaron para resolver nuestro caso de estudio y hacer las pertinentes comparaciones, que nos permitan mostrar el desempeño de campo medio.

Para validar el desempeño de la teoría de campo medio con otra heurística, se incorpora a éste una búsqueda local simple. Llamando a esta combinación de metodologías híbrido de campo medio más una búsqueda local, el cual es comparado con una estrategia de la combinación de una heurística voraz (greedy) y una búsqueda local simple. Estrategia que es bien sabido, brinda buenos resultados para cierto tamaño de casos.

Nuestro trabajo esta basado en una investigación previa [], en la que se observó que los resultados brindados por campo medio eran buenos incluso mejores que los del algoritmo genético predefinido en el software R project. Además, para casos de estudio grandes CPLEX no proporciona soluciones factibles a pesar de dejarlo ejecutando varios días, a diferencia de campo medio que sí brinda al menos una solución factible. Por lo que en el presente trabajo campo medio interpretará el papel de una heurística constructiva, que será posteriormente mejorada por una búsqueda local.

El algoritmo general presentado en el trabajo [] se muestra a continuación. En el que se observó podría ser mejorado incorporando una estrategia probabilística que nos permitiece brindar soluciones guía para generar las soluciones de campo medio.

4.1 Algoritmo campo medio

---

1. Generar una solución inicial aleatoria continua $\vec{x_0}$, en un intervalo $[0, 1]$.
2. Generar una solución de campo medio $\vec{x}_{CM}$ a partir de $\vec{x_0}$ y μ, donde $\vec{x_0}$ y μ se actualizarán iterativamente.
   - 2.1 Dando valores iniciales para los multiplicadores de KKT, $\mu = \frac{v}{p}$.
   - 2.2 Resolver la ecuación 2.4.
   - 2.3 Evaluar las condiciones de KKT. Si la precisión en su realización está por encima de un valor de tolerancia predefinida, entonces actualizamos los multiplicadores de KKT y regresamos al paso 2.2. Si no finalizamos.
3. Enviar como salida el vector solución y el valor de evaluación del mismo.

El algoritmo anterior es eficiente para generar soluciones buenas en un lapso de tiempo no muy grande. Sin embargo, la calidad de la solución disminuye cuando el caso de estudio incrementa. Lo anterior es debido a que el algoritmo de campo medio parte de una solución aleatoria generada mediante una distribución uniforme, llamada solución guía. Los valores de dicha solución son continuos en el intervalo $[0, 1]$, pero como se esta trabajando con un problema binario, estos valores son redondeados a cero o uno. Realizando varias pruebas con diferentes tamaños de caso de estudio

se observó que las soluciones en en promedio tienen un 50 % de unos y un 50 % de ceros, por lo que si la solución del problema es muy densa o poco densa, la solución guía no nos ayudará para encontrar la solución del problema sino todo lo contrario, impedirá el hallazgo de una solución de campo medio factible. Para corregir lo anteriormente mencionado, se propone, en un nuevo algoritmo, generar soluciones guía tomando en cuenta la probabilidad de ocurrencia de unos y ceros. Si las soluciones generadas por dicha probabilidad son factibles incrementamos un pequeño porcentaje la probabilidad de unos, de otra manera disminuimos el porcentaje de probabilidad de unos. Esto se realiza con la finalidad de estimar la densidad de la solución del problema, con la esperanza de encontrar la solución óptima.

4.2 Algoritmo campo medio mejorado

---

1. Generar una solución inicial aleatoria continua $\vec{x_0}$, en un intervalo $[0, 1]$. En base a esta solución, generamos la *probabilidad_de_ocurrencia** de que en la solución aparezcan unos.

2. Generar una solución de campo medio $\vec{x}_{CM}$ a partir de $\vec{x_0}$ y μ, donde $\vec{x_0}$ y μ se actualizarán iterativamente.

   2.1 Dando valores iniciales para los multiplicadores de KKT, $\mu = \frac{v}{p}$.

   2.2 Resolver la ecuación 2.4.

   2.3 Evaluar las condiciones de KKT. Si la precisión en su realización está por encima de un valor de tolerancia predefinida, entonces actualizamos los multiplicadores de KKT y regresamos al paso 2.2. Si no finalizamos.

3. Enviar como salida el vector solución y el valor de evaluación del mismo.

Nota: La *probabilidad_de_ocurrencia* cambiará iterativamente hasta que ésta encuentre un estándar de no más cambios.

El propósito del presente trabajo es validar el desempeño de campo medio más una búsqueda local simple, el cual será comparado con con un híbrido de un algoritmo voraz más una búsqueda local simple, esto debido a que se observó en la literatura que voraz proporciona soluciones muy buenas en poco tiempo [16]. Como campo medio genera una solución en base a soluciones pasadas o una solución guía apropiada, podemos estimar la probabilidad de ocurrencia de cada una de las variables de decisión, y en base a ella fijar ciertas variables que no podrán ser modificadas dentro de la búsqueda local. Esto se ejemplifica en la tabla 4.1, donde se muestran 5 pequeñas soluciones binarias generadas por campo medio. La probabilidad de ocurrencia de cada una de las variables de decisión se muestra en la fila promedio, por lo que fijamos las variables que siempre aparecen en las soluciones o bien, aquellas variables que tengan una probabilidad de ocurrencia mayor a un nivel de tolerancia permisible.

Tabla 4.1: Probabilidad de ocurrencia de las variables de decisión

	Secuencia									
Solución	1	2	3	4	5	6	7	8	9	10
1	1	1	1	0	1	1	0	1	0	1
2	1	1	0	1	1	1	1	0	1	1
3	1	1	0	1	1	1	1	0	1	1
4	1	1	1	1	1	1	0	1	1	1
5	1	1	1	0	1	1	0	1	1	1
promedio	1	1	0.6	0.6	1	1	0.4	0.6	0.8	1

Los siguientes dos algoritmo muestran la estructura básica de los híbrídos campo medio más una búsqueda local y voraz más una búsqueda local que fueron implementados para validar el desempeño del primero.

4.3 Algoritmo híbrido campo medio aunado a una búsqueda local simple

---

1. Generar una solución de campo medio $\vec{x}_{CM}$ con el algoritmo 4.2.
2. Definir un conjunto de variables fijas, las cuales siempre tomarán el valor de uno, en base a la probabilidad de ocurrencia de las soluciones generadas por campo medio.
3. Dada una solución $\vec{x}_{CM}$, definimos un vecindario $N(\vec{x}_{CM})$ en base al movimiento mencionado anteriormente en la sección 2.2 y una función $f(\vec{x}_{CM})$ a maximizar. Dicho movimiento no contempla las variables fijas.
4. Realizar la búsqueda local.

   4.1 $\vec{x} = \vec{x}_{CM}$.

   4.2 Mientras exista $\vec{y} \in N(\vec{x})$ tal que $f(\vec{y}) > f(\vec{x})$, entonces $\vec{x} = \vec{y}$.
5. Enviar como salida el vector solución y el valor de evaluación del mismo en la función objetivo.

De manera análoga al método anterior se desarrolla el algoritmo híbrido para voraz (Greedy) más una búsqueda local.

4.4 Algoritmo híbrido voraz aunado a una búsqueda local simple

---

1. Generar una solución voraz $\vec{x_v}$ con el algoritmo **??**.
2. Definir un conjunto de variables fijas, las cuales siempre tomarán el valor de uno, en base a la probabilidad de ocurrencia de las soluciones generadas por campo medio.
3. Dada una solución $\vec{x}_v$, definimos un vecindario $N(\vec{x}_v)$ en base al movimiento mencionado anteriormente en la sección 2.2 y una función $f(\vec{x}_v)$ a maximizar. Dicho movimiento no contempla las variables fijas.
4. Realizar la búsqueda local.

   4.1 $\vec{x} = \vec{x_v}$.

   4.2 Mientras exista $\vec{y} \in N(\vec{x})$ tal que $f(\vec{y}) > f(\vec{x})$, entonces $\vec{x} = \vec{y}$.
5. Enviar como salida el vector solución y el valor de evaluación del mismo en la función objetivo.

Capítulo 5

Experimentación

Algunos de los casos de estudio del problema de la mochila cuadrática que se muestran a continuación, son casos benchmark reportados en la literatura y otros son creados usando el generador de Pisinger. La experimentación fue realizada en una Laptop HP Pavilion dv4, con procesador 4x Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz, memoria de 4075 MB y sistema operativo Linux Mint 14 (32-bit). Además, la codificación de las heurísticas CM, Voraz, CMBL y VBL fueron desarrolladas en el software libre R. Para cada uno de los casos se realizaron 10 pruebas independientes para los heurísticos CM y CMBL, para validar la robustez del algoritmo de campo medio. El algoritmo de voraz que se implementa es el más simple que se ha reportado, el cual genera la misma solución si se corriera varias veces por lo que tanto para el voraz puro como al hibridado con la búsqueda local se ejecuta una sóla vez.

Los casos de estudio benchmark, los cuales son muy usados para validar la efectividad de nuevas estrategias heurísticas, son de tamaño 200 con 4 tipo de densidades. Estas densidades son de un 25 %, 50 %, 75 % y un 100 %, es decir, los porcentajes indican la fuerza de conectividad de un elemento con el resto de los elementos. Cada una de estas densidades tienen 10 casos de estudio diferentes, por lo que para mostrar el comportamiento de las heurísticas CMBL y VBL en este tamaño de caso, se promedian los GAPs y los tiempos de ejecución. Los resultados promedio obtenidos para cada una de estas 4 densidades se muestran en las figuras 5.1 y 5.2, donde 1, 2, 3 y 4 son las densidades 25 %, 50 %, 75 % y 100 %, respectivamente.

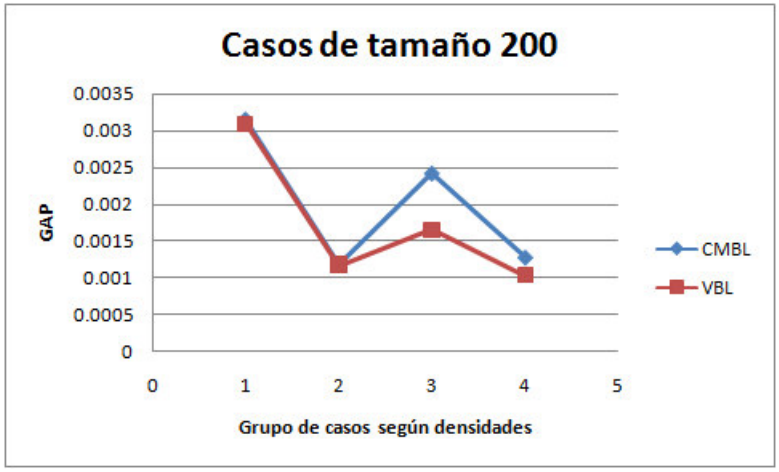


Figura 5.1: Comparativo de GAP promedio entre CMBL y VBL en casos de tamaño 200.

En la gráfica 5.1 se puede observar claramente que los GAPs promedio de las densidades son muy parecidos, aunque ligeramente la heurística VBL ofrece un GAP menor al de CMBL. Sin embargo, la diferencia entre los tiempos promedio de ambas heurísticos es grande como podemos ver en la figura 5.2. Por lo tanto, para estos casos de tamaño 200 podemos concluir que aunque los GAPs promedio son muy similares pero, el costo computacional es significativamente mayor en CMBL, por ello la heurística CMBL no es muy útil para este tipo de tamaño.

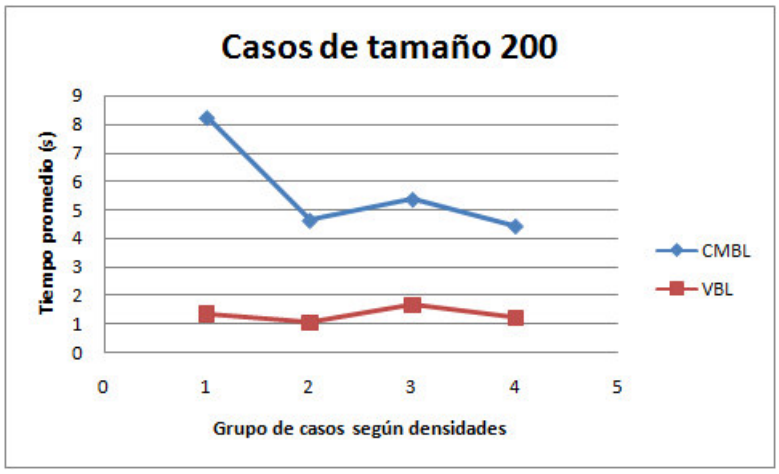


Figura 5.2: Comparativo del tiempo promedio de ejecución entre CMBL y VBL en casos de tamaño 200.

Debido a que se observó un buen desempeño en cuanto a la calidad de la solución obtenida en los casos benchmark de tamaño 200 por la heurística CMBL, se decide experimentar con casos de mayor tamaño, con la esperanza de que esta heurística sea de utilidad en casos grandes. Se crearon casos de tamaño 1000, 2000, 5000, 10000 y 20000 usando el generador de Pisinger. Como los casos son creados desde cero, se desconoce la solución óptima de cada uno de ellos, por lo que para validar la calidad de la solución se comparan las soluciones de los dos métodos, CMBL y VBL, mostrando un porcentaje de que tan mejor es uno del otro. Para los tamaños 5000, 10000 y 20000 se compara únicamente las soluciones de las heurísticas puras CM y Voraz, puesto que los tiempos de cómputo son muy elevados al realizar una búsqueda local simple.

En los casos de tamaño 1000 se puede observar que ligeramente el método de CMBL brinda un porcentaje mayor en cuanto a la calidad de la solución, como se puede apreciar en la gráfica 5.3. Así mismo, los tiempos de ejecución son muy parecidos e incluso menores en ciertos casos, ver la gráfica 5.4.

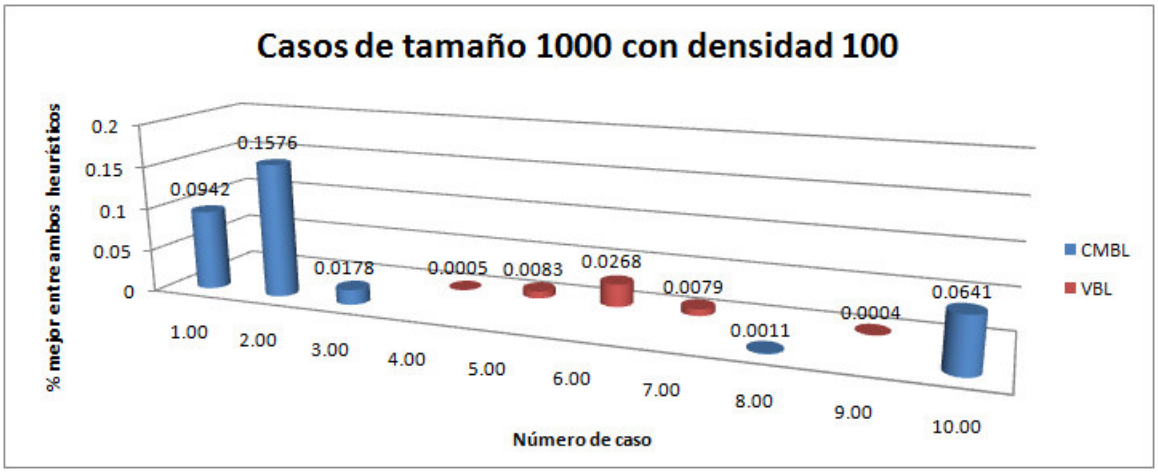


Figura 5.3: Comparativo del mejor porcentaje del valor de la función objetivo entre CMBL y VBL, en casos de tamaño 1000 con una densidad del 100 %.

Al observar los resultados obtenidos en los casos de tamaño 1000, ver que la calidad de la solución de CMBL es muy parecida al de VBL e inclusive mejor y que además los tiempos de respuesta cada vez son menores para CMBL, esto debido

a que la heurística de campo medio pura, permite fijar ciertas variables en base a su probabilidad de aparición, a diferencia de la heurística Voraz quién únicamente incorpora elementos a "ciegas ". Por esta razón, suponemos que la heurística de campo medio es una buena estrategia para tamaño de gran escala y que puede brindar soluciones de buena calidad en un tiempo razonable de cómputo, todo ello con ayuda de alguna otra estrategia.

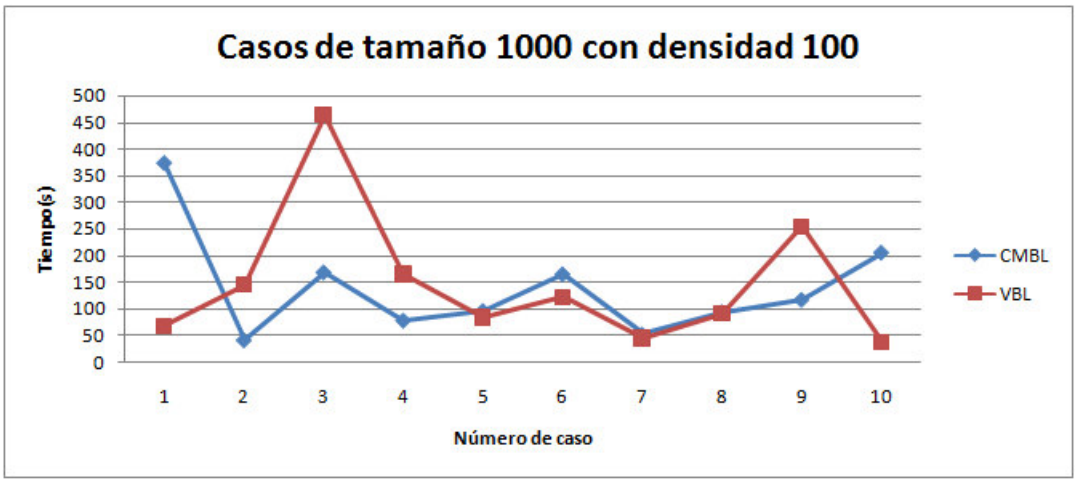


Figura 5.4: Comparativo del tiempo promedio de ejecución entre CMVBL y VBL, en casos de tamaño 1000.

A continuación se muestran los resultados obtenidos para casos de tamaño 2000, en la gráfica 5.5 se puede observar que los porcentajes de la calidad de la solución entre los métodos CMBL y VBL se siguen manteniendo muy similares. Sin embargo, en la gráfica 5.6 se comienza a notar una diferencia significativa entre los tiempos de respuesta de ambos métodos, la heurística CMBL da respuesta en un tiempo menor que la heurística VBL. Por lo tanto, la heurística CMBL brinda soluciones muy similares a las obtenidas por la otra, en un tiempo significativamente menor por lo que se esperaría que para tamaños más grandes, CMBL proporcione soluciones buenas en un menor tiempo al de VBL.

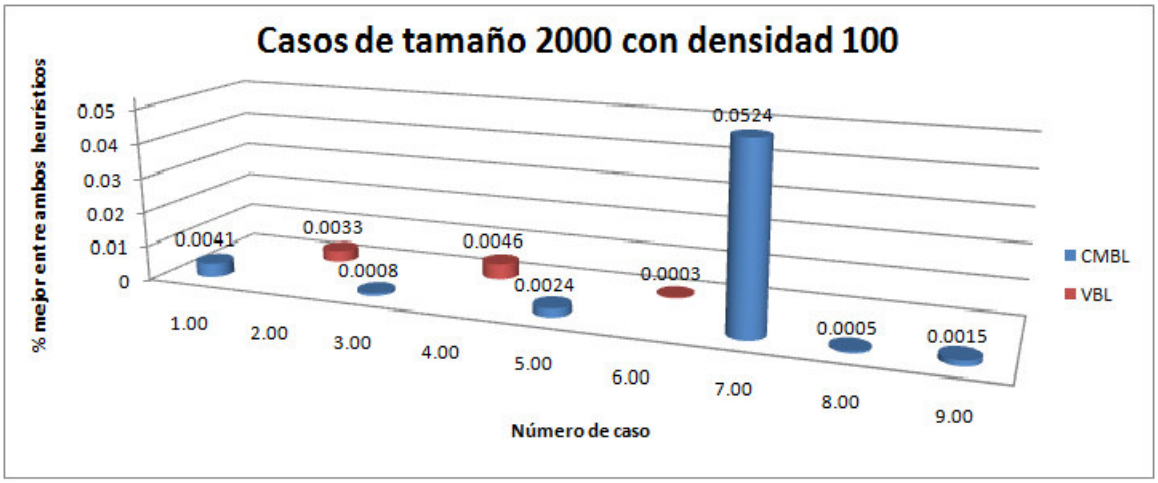


Figura 5.5: Comparativo del mejor porcentaje del valor de la función objetivo entre CMBL y VBL, en casos de tamaño 2000 con una densidad del 100 %

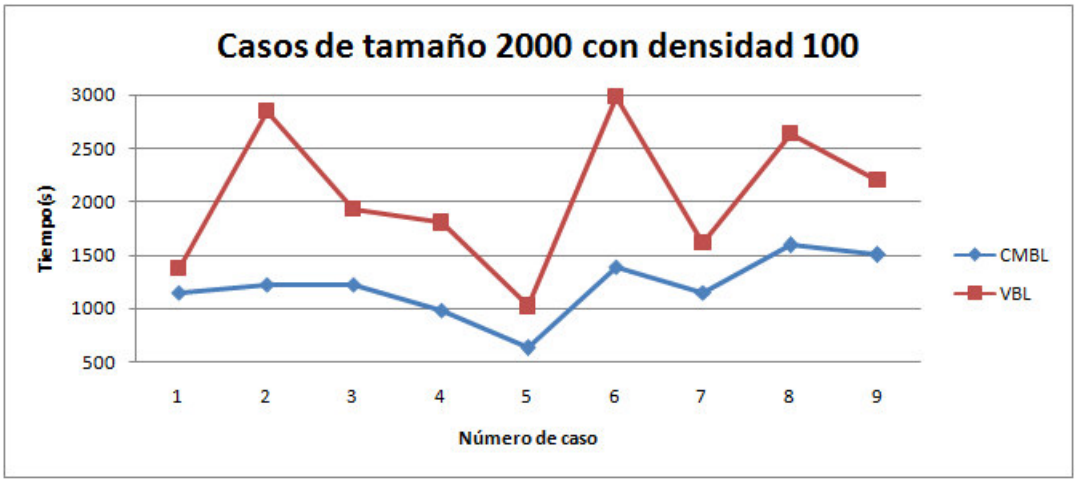


Figura 5.6: Comparativo del tiempo promedio de ejecución entre CMBL y VBL, en casos de tamaño 2000.

Para casos de tamaño 5000 no fue posible realizar una búsqueda local simple debido a que el costo computacional era muy alto, por ende para los casos de 10000 y 20000 tampoco se realizó. Por ello, enseguida se muestran únicamente los resultados obtenidos por las heurísticas puras, CM y Voraz. En cada uno de los tamaños se observó que el método voraz ofrece respuesta en un tiempo mucho menor como era de esperarse, esto debido a que el método voraz incorpora elementos a la solución

sin tomar encuenta la estructura del problema, simplemente obedece a una función de evaluación bien definida y que se sabe, existen varias que generan soluciones de muy buena calidad para el problema de la mochila.

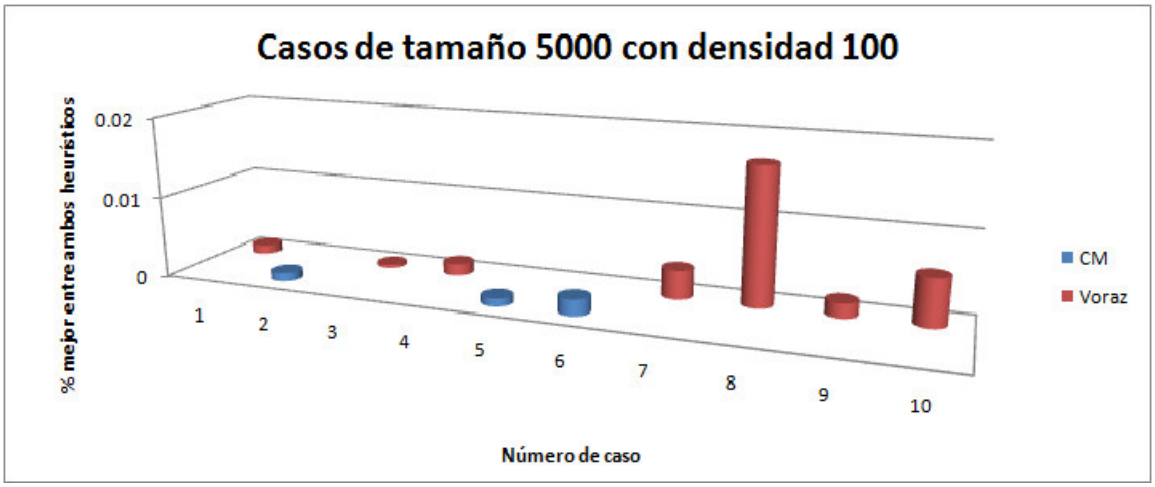


Figura 5.7: Comparativo del mejor porcentaje del valor de la función objetivo entre CM y Voraz, en casos de tamaño 5000 con una densidad del 100 %

En las gráficas 5.7, 5.8 y 5.9, podemos observar que en el porcentaje de la calidad de la solución para la mayoría de los casos el método voraz tiene la ventaja, puesto que ofrece un porcentaje significativamente mayor en la calidad de las soluciones. Por ello, podemos pensar en que el método de campo medio por sí solo en instancias grandes no es la metodología indicada para el problema de la mochila cuadrática.

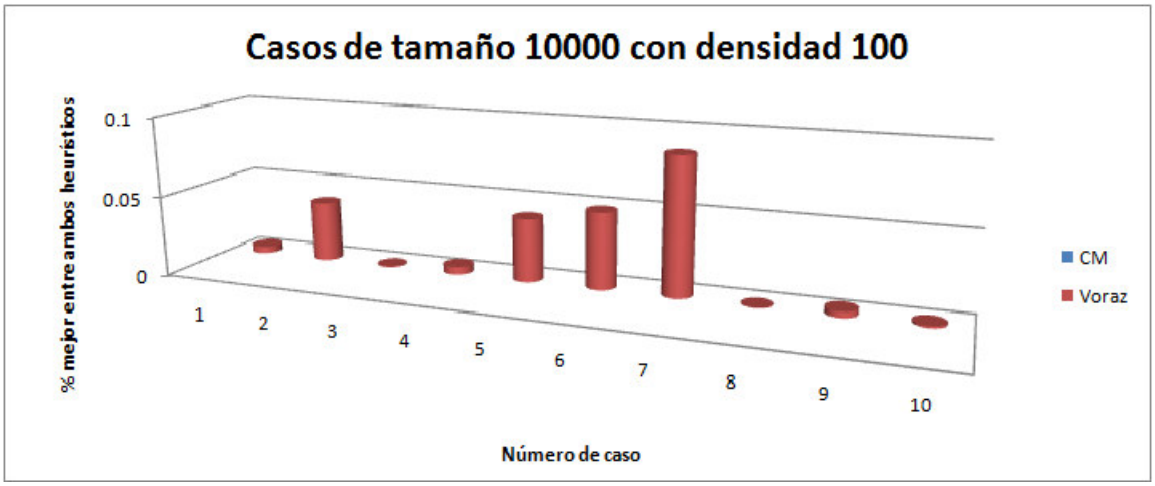


Figura 5.8: Comparativo del mejor porcentaje del valor de la función objetivo entre CM y Voraz, en casos de tamaño 10000 con una densidad del 100 %

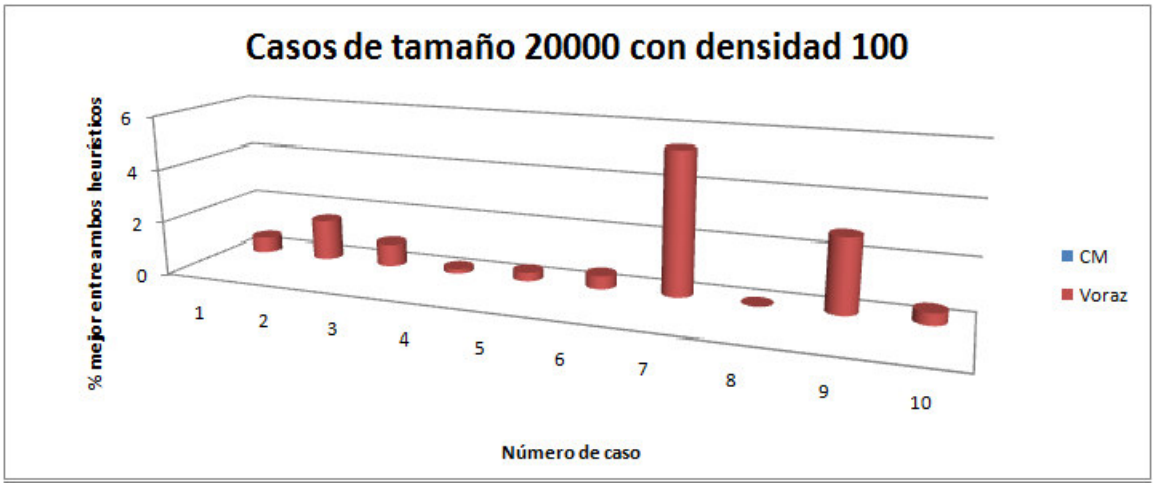


Figura 5.9: Comparativo del mejor porcentaje del valor de la función objetivo entre CM y Voraz, en casos de tamaño 20000 con una densidad del 100 %

Para mayor información sobre los resultados obtenidos, ver las tablas que se encuentran en el apéndice de los diferentes casos de estudio de la experimentación. Así mismo, cabe mencionar que los datos para realizar cada una de las graficas fueron tomados de estas mismas tablas.

Capítulo 6

Conclusiones y trabajo futuro

Se sabe que el algoritmo voraz es una muy buena metodología para resolver el problema de la mochila y sus derivados, por lo que el competir con esta y observar resultados similares e inclusive mejores es un gran aporte. Las soluciones generadas por la heurística de campo medio por si solo son de buena calidad como lo pudimos observar en los casos de 5000, 10000 y 20000. Además, incorporando una búsqueda local simple las soluciones de ambas heurísticas son mejoradas y éstas siguen siendo competitivas pero, los tiempos en tamaños de 1000 y 2000 campo medio puede ofrecer soluciones en un tiempo significativamente menor al de voraz. Por lo que se esperaría que en casos más grandes campo medio siga brindando soluciones similares al voraz en un menor tiempo. Sin embargo, esta hipótesis no pudo ser comprobada puesto que en tamaños de mayor magnitud la búsqueda local es ineficiente ya que el tiempo de respuesta es muy alto. Por esta razón, se piensa que tal vez la búsqueda local simple no fue la mejor opción para realizar las mejoras de la solución. Más sin embargo, nos ayudo a darnos cuenta que las variables que campo medio identifica como fijas, no afectan en la mejora de la solución sino todo lo contrario, ayudan a disminuir el espacio de búsqueda y por ende el tiempo de respuesta.

Como trabajo futuro se quiere incorporar otro tipo de estrategia que mejore las soluciones en tamaños superiores a los 5000 en el menor tiempo posible. Además, validar el desempeño de campo medio en otros tipos de problemas combinatorios de interés.

Apéndice A

Apéndice

		Constructivo puro			Constructivo con búsqueda local			
Caso	Óptimo	Constructivo	Valor prom.	t(s)	Valor prom.	Destd	t(s)	GAP
1	204441	CM	202619.9	2.13	204101.4	183.09	3.038	0.00166
		Voraz	203940	0.012	204102		0.5	0.00166
2	239573	CM	238597.8	6.8	239447.1	92.54	7.035	0.00053
		Voraz	239296	0.008	239573		0.22	0
3	245463	CM						
		Voraz						
4	222361	CM	220728.3	7.67	221970.3	231.22	8.311	0.00176
		Voraz	222248	0.008	222248		0.19	0.00051
5	187324	CM	185229.3	3.6	187005.7	159.06	5.168	0.0017
		Voraz	185232	0.012	187129		1.37	0.00104
6	80351	CM	79534.3	9.23	80007.1	106.88	9.446	0.00428
		Voraz	78028	0.012	80237		2.33	0.00142
7	59036	CM	58263.2	6.68	58618.5	156.38	6.944	0.00707
		Voraz	57884	0.008	58665		2.25	0.00628
8	149433	CM	148319.3	6.29	149175.5	107.1	6.884	0.00172
		Voraz	147990	0.012	149324		1.06	0.00073
9	49366	CM	48860.1	9.77	49075.3	259.44	9.962	0.00589
		Voraz	48293	0.008	49123		1.62	0.00492
10	48459	CM	48161.8	17.04	48276.1	259.44	17.345	0.00377
		Voraz	46109	0.012	47916		2.71	0.01121

Tabla A.1: Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 25 %.

		Constructivo puro			Constructivo con búsqueda local			
Caso	Óptimo	Constructivo	Valor prom.	t(s)	Valor prom.	Destd	t(s)	GAP
1	372097	CM	370576.2	2.34	371578.4	272.89	3.025	0.00139
		Voraz	371393	0.008	371722		0.66	0.00101
2	211130	CM	209425.9	3.67	210857	182.41	4.183	0.00129
		Voraz	209332	0.008	210762		0.76	0.00174
3	227185	CM	225379	8.27	226518.1	371.08	8.864	0.00294
		Voraz	223479	0.012	226460		1.85	0.00319
4	228572	CM	228127.5	8.5	228466.5	173.03	8.732	0.00046
		Voraz	227751	0.008	228572		1.71	0
5	479651	CM	478088.7	2.76	479419.5	207.74	3.084	0.00048
		Voraz	479004	0.008	479451		0.32	0.00042
6	426777	CM	424346.7	2.52	426408.3	710.57	2.928	0.00086
		Voraz	424893	0.008	426488		0.28	0.00068
7	220890	CM	219256.6	2.17	220561.1	423.39	2.901	0.00149
		Voraz	218324	0.012	220277		1.4	0.00278
8	317952	CM	316736.1	2.46	317616.8	198.11	3.312	0.00105
		Voraz	316732	0.012	317880		1.07	0.00023
9	104936	CM	103869.6	5.39	104845.1	157.71	5.991	0.00087
		Voraz	103494	0.012	104936		1.53	0
10	284751	CM	283023.2	2.14	284474.5	253.57	3.055	0.00097
		Voraz	283193	0.012	284269		1.03	0.00169

Tabla A.2: Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 50 %.

		Constructivo puro			Constructivo con búsqueda local			
Caso	Óptimo	Constructivo	Valor prom.	t(s)	Valor prom.	Destd	t(s)	GAP
1	442894	CM	440105.4	0.62	442441	397.14	1.856	0.00102
		Voraz	437929	0.008	442434		1.89	0.00104
2	286643	CM	283958.3	6.72	286066.9	951.86	7.78	0.00201
		Voraz	283839	0.008	286582		2.04	0.00021
3	61924	CM	60732	15.4	61258.6	320.21	15.55	0.01075
		Voraz	60698	0.012	61661		1.51	0.00425
4	128351	CM	127640.1	3.43	128065.7	277.45	3.971	0.00222
		Voraz	126909	0.008	128351		2.91	0
5	137885	CM	136079.1	5.72	137629.8	290.42	6.458	0.00185
		Voraz	134462	0.008	137764		3.12	0.00088
6	229631	CM	229047	3.7	229547.4	36.04	4.225	0.00036
		Voraz	228765	0.012	229493		0.96	0.00060
7	269887	CM	269025.2	7.24	269750.1	117.97	7.517	0.00051
		Voraz	267580	0.012	269648		1.54	0.00089
8	600858	CM	598077.5	1.07	599748.2	1310.72	1.677	0.00185
		Voraz	600631	0.008	600711		0.36	0.00024
9	516771	CM	514779.4	0.89	516547.3	190.43	1.475	0.00043
		Voraz	514428	0.012	516599		1.48	0.00033
10	142694	CM	140911.4	2.55	142243.8	475.73	3.006	0.00316
		Voraz	139787	0.008	141547		0.87	0.00804

Tabla A.3: Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 75 %..

		Constructivo puro			Constructivo con búsqueda local			
Caso	Óptimo	Constructivo	Valor prom.	t(s)	Valor prom.	Destd	t(s)	GAP
1	937149	CM	935322.4	2.15	937069.5	83.8	2.543	0.00008
		Voraz	935700	0.008	937149		0.69	0
2	303058	CM	300553.7	3.07	302778	485.05	3.488	0.00092
		Voraz	300427	0.008	303058		1.78	0
3	29367	CM	28873.4	17.05	29110.3	46.49	17.302	0.00874
		Voraz	28455	0.008	29125		1.16	0.00824
4	100838	CM	100183.3	4.13	100799.2	117.18	4.344	0.00038
		Voraz	99610	0.008	100838		1.02	0
5	786635	CM	785139.8	0.59	786590.7	71.36	1.143	0.00006
		Voraz	785441	0.008	786635		0.62	0
6	41171	CM	41171	5.37	41171	0	5.491	0
		Voraz	41171	0.012	41171		0.34	0
7	701094	CM	698608.9	0.39	700460.3	366.73	1.051	0.00090
		Voraz	698566	0.008	700417		1.2	0.00097
8	782443	CM	779034.9	0.76	781800.2	44.19	1.506	0.00082
		Voraz	779304	0.008	781780		0.82	0.00085
9	628992	CM	627085.3	0.89	628750.4	255	1.989	0.00038
		Voraz	626668	0.012	628992		1.05	0
10	378442	CM	376394.1	4.17	378234.4	96.3	5.192	0.00055
		Voraz	376219	0.008	378341		3.43	0.00027

Tabla A.4: Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 200 con una densidad del 100 %.

	Constructivo puro			Constructivo con búsqueda local			% Mejor	
Caso	Constructivo	Valor prom.	t(s)	Valor prom.	Destd	t(s)	CMBL	VBL
1	CM	23592112.2	16.089	23622743.8	42885.85	373.055	0.0942	
	Voraz	23625965	0.088	23645022		67.044		
2	CM	1055031.2	13.478	1058719.3	1327.94	41.473	0.1576	
	Voraz	1054584	0.1	1060391		144.845		
3	CM	6606146.9	4.4	6618808.7	483.02	168.581	0.0178	
	Voraz	6605051	0.092	6619987		462.265		
4	CM	7042198.5	4.247	7044817.9	110.36	78.26		0.0005
	Voraz	7041271	0.088	7044783		165.638		
5	CM	20841139.7	6.328	20858243	714.22	96.365		0.0083
	Voraz	20839896	0.072	20856518		84.741		
6	CM	16117471.4	4.814	16133270.5	1387.86	164.978		0.0268
	Voraz	16112519	0.1	16128941		122.047		
7	CM	15531649.9	3.401	15538512.4	1546.91	52.048		0.0079
	Voraz	15533569	0.084	15537282		45.427		
8	CM	18884148.3	7.899	18892659.3	795.9	94.066	0.0011	
	Voraz	18883185	0.092	18892869		91.053		
9	CM	6102032.8	3.919	6108334.1	941.77	117.165		0.0004
	Voraz	6100889	0.1	6108307		253.544		
10	CM	22316101.8	9.755	22329437.2	44944.68	205.436	0.0641	
	Voraz	22335630	0.096	22343765		38.466		

Tabla A.5: Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 1000 con una densidad del 100 %.

	Constructivo puro			Constructivo con búsqueda local			% Mejor	
Caso	Constructivo	Valor prom.	t(s)	Valor prom.	Destd	t(s)	CMBL	VBL
1	CM	58626245	35.066	58643650	0	1147.78	0.0041	
	Voraz	58619422	0.296	58641259		1376.998		
2	CM	27196170.4	8.557	27223370.6	2638.41	1221.601		0.0033
	Voraz	27192994	0.316	27224273		2843.953		
3	CM	42014364	8.05	42034421.7	1183.78	1224.463	0.0008	
	Voraz	42016063	0.304	42034096		1929.152		
4	CM	55598015.4	9.454	55616518.8	2357.3	982.372		0.0046
	Voraz	55600707	0.272	55619090		1800.652		
5	CM	59358500.3	9.746	59375682	750.51	627.543	0.0024	
	Voraz	59351047	0.3	59374241		1019.619		
6	CM	35890404.4	13.24	35913346	1659.74	1385.359		0.0003
	Voraz	35890149	0.264	35913454		2980.49		
7	CM	24298887.8	9.374	24322099.9	1023.92	1150.449	0.0524	
	Voraz	24297978	0.317	24309351		1614.861		
8	CM	46102211.6	16.536	46130397.9	1780.76	1588.688	0.0005	
	Voraz	46105314	0.288	46130174		2632.312		
9	CM	70245821.6	17.868	70273371.6	3101.63	1501.109	0.0015	
	Voraz	70250252	0.248	70272294		2196.766		

Tabla A.6: Resultados experimentales obtenidos de CMBL y VBL, en tamaños de casos 2000 con una densidad del 100 %.

	Campo medio puro			Voraz puro		% Mejor	
Caso	Valor prom.	Destd	t(s)	Valor prom.	t(s)	CM	Voraz
1	517203430.20	2468.68	61.62	517209155.00	1.19		0.00111
2	332417340.40	5574.59	17.07	332413939.00	1.20	0.00102	
3	596666118.00	1777.93	62.34	596668639.00	1.23		0.00042
4	501015935.70	6299.77	56.55	501023327.00	1.26		0.00148
5	304221068.20	15341.17	15.60	304218167.00	1.21	0.00095	
6	163482381.50	11847.39	18.96	163479192.00	1.35	0.00195	
7	189678997.30	11178.79	16.64	189685814.00	1.20		0.00359
8	42593086.40	4101.55	17.19	42600231.00	1.26		0.01677
9	398435607.70	5333.14	33.06	398442587.00	1.20		0.00175
10	570879969.20	81082.30	127.06	570912855.00	1.27		0.00576

Tabla A.7: Resultados experimentales obtenidos de CM y Voraz, en tamaños de casos 5000 con una densidad del 100 %.

	Campo medio puro			Voraz puro		% Mejor	
Caso	Valor prom.	Destd	t(s)	Valor prom.	t(s)	CM	Voraz
1	154687845.30	4731.26	100.53	154694499.00	4.73		0.00430
2	1872852374.00	1204594.24	331.10	1873565939.00	4.60		0.03809
3	1613736284.30	4791.05	208.52	1613745511.00	4.65		0.00057
4	2540226397.10	175342.92	381.21	2540354154.00	4.77		0.00503
5	832674335.90	1039136.97	146.64	833007137.00	4.68		0.03995
6	500357043.50	658732.53	132.71	500598504.00	4.76		0.04823
7	1260042643.80	3420439.67	95.78	1261129012.00	4.92		0.08614
8	911639500.00	7300.33	89.78	911641974.00	4.78		0.00027
9	648031843.20	10519.76	52.36	648055733.00	4.70		0.00369
10	1443285485.80	10590.36	95.91	1443304299.00	4.62		0.00130

Tabla A.8: Resultados experimentales obtenidos de CM y Voraz, en tamaños de casos 10000 con una densidad del 100 %.

	Campo medio puro			Voraz puro		% Mejor	
Caso	Valor prom.	Destd	t(s)	Valor prom.	t(s)	CM	Voraz
1	2379750681.30	11939382.40	379.71	2393224585.00	21.35		0.56300
2	9419100496.20	59179973.83	412.06	9563918942.00	22.40		1.51422
3	5025563938.30	72188647.72	329.45	5068888270.00	20.63		0.85471
4	4074317811.20	22258045.87	260.04	4081397335.00	24.71		0.17346
5	1663137102.70	6512671.24	351.62	1668710502.00	24.59		0.33399
6	4267865945.10	39474980.52	358.46	4289410682.00	21.57		0.50228
7	7315429078.20	2033756194.03	407.82	7726734855.00	23.20		5.32315
8	4894819841.20	100202.67	278.09	4894821316.00	22.27		0.00003
9	3088136665.50	84568100.34	341.99	3175488633.00	22.93		2.75082
10	2031144102.30	19119736.58	312.46	2040256891.00	21.76		0.44665

Tabla A.9: Resultados experimentales obtenidos de CM y Voraz, en tamaños de casos 20000 con una densidad del 100 %.

Bibliografía

[1] Aarts, E. H. y J. K. Lenstra, *Local search in combinatorial optimization*, Princeton University Press, 2003.

[2] Anderson, E., «Mechanisms for local search», *European Journal of Operational Research*, **88**(1), págs. 139–151, 1996.

[3] Bednorz, W., «Advances in greedy algorithms», *Wienna: I-Tech Education and Publishing KG*, 2008.

[4] Bellmore, M. y G. L. Nemhauser, «The traveling salesman problem: a survey», *Operations Research*, **16**(3), págs. 538–558, 1968.

[5] Bishop, C. M. y N. M. Nasrabadi, *Pattern recognition and machine learning*, tomo 1, springer New York, 2006.

[6] Blum, C., J. Puchinger, G. R. Raidl y A. Roli, «Hybrid metaheuristics in combinatorial optimization: A survey», *Applied Soft Computing*, **11**(6), págs. 4135–4151, 2011.

[7] Blum, C. y A. Roli, «Metaheuristics in combinatorial optimization: Overview and conceptual comparison», *ACM Computing Surveys (CSUR)*, **35**(3), págs. 268–308, 2003.

[8] Blum, C. y A. Roli, «Metaheuristics in combinatorial optimization: Overview and conceptual comparison», *ACM Comput. Surv.*, **35**(3), págs. 268–308, septiembre 2003, URL `http://doi.acm.org/10.1145/937503.937505`.

[9] Blum, C. y A. Roli, «Hybrid metaheuristics: an introduction», en *Hybrid Metaheuristics*, Springer, págs. 1–30, 2008.

[10] Brassard, G. y P. Bratley, *Fundamentals of algorithmics*, tomo 524, Prentice Hall Englewood Cliffs, 1996.

[11] Castillo, E., A. J. Conejo, P. Pedregal, R. Garcia y N. Alguacil, «Formulación y Resolución de Modelos de Programación Matemática en Ingenieria y Ciencia.», , 2002.

[12] Fernández, A. D., J. G. Velarde, M. Laguna, P. Mascato, F. Tseng, F. Glover y H. Ghaziri, «Optimización heurística y redes neuronales», *ed. Paraninfo SA, Madrid, España*, 1996.

[13] Gendreau, M. y J.-Y. Potvin, *Handbook of metaheuristics*, tomo 146, Springer, 2010.

[14] Goyal, S., «A Survey on Travelling Salesman Problem», en *Proc. 43rd Midwest Instruction and Computing Symposium (MICS 2010)*, 2010.

[15] Hoos, H. H. y T. Stützle, *Stochastic local search: Foundations and applications*, Access Online via Elsevier, 2004.

[16] Julstrom, B. A., «Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem», en *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, ACM, New York, NY, USA, págs. 607–614, 2005, URL `http://doi.acm.org/10.1145/1068009.1068111`.

[17] Laporte, G., «The vehicle routing problem: An overview of exact and approximate algorithms», *European Journal of Operational Research*, **59**(3), págs. 345–358, 1992.

[18] Lawler, E. L. y D. E. Wood, «Branch-and-bound methods: A survey», *Operations research*, **14**(4), págs. 699–719, 1966.

[19] Luke, S., *Essentials of Metaheuristics: A Set of Undergraduate Lecture Notes by*, Lulu. com, 2009.

[20] Martí, R., «Procedimientos Metaheurísticos en Optimización Combinatoria», *Informe técnico*, Departmento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universidad de Valencia.

[21] Mitchell, J. E., «Branch and cut», *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[22] Nemhauser, G. L. y L. A. Wolsey, *Integer and combinatorial optimization*, tomo 18, Wiley New York, 1988.

[23] Neumann, F. y C. Witt, *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*, Springer, 2010.

[24] Ohlsson, M., C. Peterson y B. Söderberg, «An efficient mean field approach to the set covering problem», *European Journal of Operational Research*, **133**(3), págs. 583–595, 2001.

[25] Ohlsson, M. y H. Pi, «A study of the mean field approach to knapsack problems», *Neural Networks*, **10**(2), págs. 263–271, 1997.

[26] Opper, M. y D. Saad, *Advanced mean field methods: Theory and practice*, the MIT press, 2001.

[27] Osman, I. H. y G. Laporte, «Metaheuristics: A bibliography», *Annals of Operations Research*, **63**(5), págs. 511–623, 1996.

[28] Papadimitriou, C. H. y K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, Courier Dover Publications, 1998.

[29] Pelikan, M., D. Goldberg y F. Lobo, «A Survey of Optimization by Building and Using Probabilistic Models», *Computational Optimization and Applications*, **21**(1), págs. 5–20, 2002, URL `http://dx.doi.org/10.1023/A%3A1013500812258`.

[30] Pelikan, M., K. Sastry y E. Cantú-Paz, *Scalable optimization via probabilistic modeling: From algorithms to applications*, tomo 33, Springer, 2006.

[31] Pinedo, M., *Scheduling: theory, algorithms, and systems*, Springer, 2012.

[32] Pisinger, D., «The quadratic knapsack problem: a survey», *Discrete Applied Mathematics*, **155**(5), págs. 623 – 648, 2007, URL `http://www.sciencedirect.com/science/article/pii/S0166218X06003878`.

[33] Rolland, E., D. A. Schilling y J. R. Current, «An efficient tabu search procedure for the p-median problem», *European Journal of Operational Research*, **96**(2), págs. 329–342, 1997.

[34] Salkin, H. M. y C. A. De Kluyver, «The knapsack problem: a survey», *Naval Research Logistics Quarterly*, **22**(1), págs. 127–144, 1975.

[35] Stützle, T. G., *Local search algorithms for combinatorial problems: analysis, improvements, and new applications*, tomo 220, Infix Sankt Augustin, Germany, 1999.

[36] Talbi, E.-G., *Metaheuristics: from design to implementation*, tomo 74, John Wiley and Sons, 2009.

[37] Toth, P. y D. Vigo, *The vehicle routing problem*, Siam, 2001.

[38] Xie, X.-F. y J. Liu, «A Mini-Swarm for the Quadratic Knapsack Problem», en *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, págs. 190–197, 2007.

[39] Yang, Z., G. Wang y F. Chu, «An effective GRASP and tabu search for the 0 1 quadratic knapsack problem», *Computers and Operations Research*, **40**(5), págs. 1176 – 1185, 2013, URL `http://www.sciencedirect.com/science/article/pii/S0305054812002675`.

Ficha autobiográfica

Juan Antonio Banda Moreno

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

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Estudio de un algoritmo heurístico híbrido basado en la teoría de campo medio aunado a una búsqueda local para el problema de la mochila cuadrática

Nací un 30 de diciembre de 1986 en la hermosa huasteca potosina, para ser más específico Cd. Valles, S.L.P. Mi educación básica y preparatoria la realicé en dicha ciudad. En agosto del 2005 decidí aventurarme a continuar mis estudios profesionales en la Universidad Autónoma de Nuevo León. En dicha institución estudié en la Facultad de Ciencias Físico Matemáticas la Licenciatura en Matemáticas, finalizando en junio de 2005. Un semestre después tomé una de las decisiones más importantes de mi vida, seguir estudiando un posgrado. Actualmente sigo siendo estudiante de maestría de tiempo completo en el Posgrado de Ingeniería de Sistemas de FIME-UANL.