FACULTAD REGIONAL MULTIDISCIPLINARIA DE CARAZO
FAREM -JINOTEPE

PROGRAMA DE DOCTORADO EN MATEMATICA APLICADA

Tesis Para Optar al Grado de Doctor en Matemática Aplicada.

ALGORITMOS HEURISTICOS EFICIENTES PARA PROBLEMAS DE CORTES ORTOGONALES EN DOS DIMENSIONES.

MÁSTER: FRANCISCO ANTONIO HERNANDEZ CRUZ.

TUTOR: Dr. RAMON ANTONIO PARAJON GUEVARA

JINOTEPE 23 DE ABRIL 2023.

AGRADECIMIENTO

Gracias señor… por llegar el final

Dedico este Tesis principalmente a Dios, por haberme dado la vida, sabiduría y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. Al Dr. Antonio Para jon por su aporte academia y su amistad, sobre todo.

A mi Esposa e hijas, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional sin importar nuestras diferencias de opiniones.

A mis padres, a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí.

FRANCISCO ANTONIO HERNANDEZ CRUZ

DEDICATORIA

Al Creador de todas las cosas Jesucristo mi padre celestial, el que me ha dado fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad que de mi corazón puede emanar, dedico primeramente mi tesis a Dios.

De igual forma, dedico esta tesis a mis padres que ha sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momento. Al Dr. Antonio Parajon Guevara por su conocimiento y experiencia, profesora Zulema Guevara, Bayardo Rosales junior gracias a su apoyo, hicieron de esta experiencia una de las más especiales en mi vida y de mi familia.

FRANCISCO HERNANDEZ CRUZ

CARTA AVAL

ANTONIO PARAJÓN GUEVARA. Profesor titular Jubilado del Departamento de Matemáticas, de la Facultad de Educación e Idiomas de la Universidad Nacional Autónoma de Nicaragua, Managua.

CERTIFICA que la presente memoria de investigación:

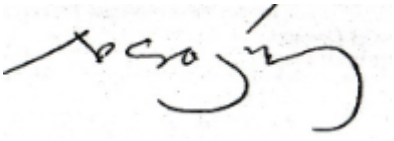
ALGORITMOS HEURÍSTICOS EFICIENTES PARA PROBLEMAS DE CORTE EN DOS DIMENSIONES

Ha sido realizado bajo su dirección en el programa de Doctorado en Matemática Aplicada, por el Ingeniero Civil: **Francisco Hernández Cruz,** y constituye su trabajo para optar al grado de Doctor en matemática aplicada.

Por lo tanto, cumple con todos los criterios y requisitos demandados por la normativa vigente de posgrado.

Managua, Nicaragua, 16 de abril de 2023.

El director de la tesis



MSc. PhD Antonio Parajon Guevara

RESUMEN

El objetivo es determinar la disposición óptima de las piezas en una lámina de tamaño fijo, de modo que se reduzca al mínimo la cantidad de material sobrante o los cortes adicionales necesarios.

Este problema es común en diversas industrias, como la manufacturera, la industria del mueble, la fabricación de textiles, entre otras. La complejidad del problema radica en que se deben considerar diferentes restricciones, como la forma y tamaño de las piezas, las restricciones de orientación, las limitaciones de espacio y la minimización del desperdicio.

Existen diferentes enfoques y algoritmos para resolver el problema de corte en dos dimensiones, que van desde métodos heurísticos hasta algoritmos exactos. Algunos de estos métodos utilizan técnicas de programación lineal, algoritmos genéticos, optimización combinatoria y enfoques basados en la metaheurística.

El objetivo final es encontrar una solución óptima que permita maximizar la utilización del material y minimizar el tiempo y los recursos necesarios para realizar los cortes. Esto se traduce en beneficios económicos, reducción de desperdicio y aumento de la eficiencia en la producción.

Algunas características importantes del problema de corte en dos dimensiones son:

1. Restricciones de forma y tamaño: Cada pieza tiene una forma y un tamaño específicos que deben ser tenidos en cuenta al realizar los cortes. Por ejemplo, las piezas pueden ser rectangulares, cuadradas, circulares u otras formas geométricas.

2. Restricciones de orientación: Dependiendo del tipo de pieza y de la aplicación, puede haber restricciones en cuanto a la orientación de las piezas en la lámina. Por ejemplo, algunas piezas pueden requerir ser colocadas en posición horizontal o vertical, o pueden tener restricciones de rotación.

3. Restricciones de cantidad: Puede haber limitaciones en la cantidad de cada tipo de pieza disponible, lo cual agrega complejidad al problema. Es necesario encontrar una disposición que respete estas restricciones de cantidad.

4. Restricciones de espacio: El espacio disponible en la lámina también puede ser limitado. Se debe asegurar que las piezas no se superpongan y que haya suficiente espacio entre ellas para los cortes.

5. Minimización del desperdicio: El objetivo principal del problema de corte en dos dimensiones es minimizar el desperdicio de material. Esto implica encontrar una solución que utilice la menor cantidad de láminas posibles y reduzca al mínimo el área no utilizada.

Para resolver el problema de corte en dos dimensiones, se han desarrollado diferentes enfoques:

a. Métodos heurísticos: Estos métodos ofrecen soluciones aproximadas, pero rápidas, al problema. Algunos ejemplos son el algoritmo de colocación primero en llegar, primero en ser servido (FFD), el algoritmo de mejor ajuste (BFD) y el algoritmo de enfoque descendente.

b. Algoritmos exactos: Estos métodos garantizan encontrar la solución óptima al problema, pero pueden requerir más tiempo computacional. Algunos enfoques utilizados incluyen programación lineal entera, programación dinámica y algoritmos de ramificación y poda.

c. Metaheurísticas: Estas técnicas buscan combinar características de diferentes enfoques para encontrar soluciones eficientes. Algunas metaheurísticas comunes son los algoritmos genéticos, el recocido simulado y la búsqueda tabú.

La elección del método a utilizar depende de la complejidad del problema, los recursos disponibles y las restricciones específicas de la aplicación. En general, el objetivo es encontrar una solución que optimice la utilización del material y minimice el desperdicio, lo que puede resultar en ahorros significativos en costos y recursos para las empresas.

INDICE GENERAL

1. INTRODUCCIÓN ---------- 1

1.1 Descripción del problema---------- 1

1.2 Justificación ---------- 6

1.3 Objetivos **----------** 10

2. REVISIÓN DEL ESTADO DEL ARTE ---------- 11

2.1 Restricciones ---------- 15

2.1.1 Cortes Ortogonales ---------- 16

2.1.2 Cortes Guillotina ---------- 17

2.1.3 Cortes en Etapas **----------** 10

2.2 Método de solución---------- 20

2.2.1 Versión no restringida del problema TDC ----- 21

2.2.2 Método exacto---------- 22

2.2.3 Hifi y Zissimopoulos**----------** 25

2.3 Método Aproximado ---------- 26

2.3.1 Morabito---------- 27

2.3.2 Fayard y Zissmopoulos ---------- 28

2.3.3 Hifi**----------** 28

2.4 Versión restringido del problema TDC----------31

2.4.1 Christofides y Whitlock---------- 32

2.4.2 Viswanathan y Bagchi---------- 34

2.4.3 Christofides y Hadjiconstantinou---------- 35

2.4.4 Hfi---------- 35

2.4.5 Hifi y Zissmopoulos**----------** 36

2.5 Métodos Aproximados----------37

2.5.1 Wang ---------- 37

2.5.2 Morabito y Aranales---------- 38
2..5.3 Fayard et. al. ---------- 39
3. ALGORITMOS HEURISTICOS ---------- 41
3.1 Algoritmos con orientación fija, rotada ------- 42
3.1.2 Cota Superior BK_j ---------- 42
3.1.3 Los procedimientos constructivos---------- 45
3.2 procedimientos constructivos Bottom Left -- 46
4**. ALGORITMO METAHEURISTICOS** ---------- 53
4.1 Definición de Movimientos---------- 53
4.1.1 Selección de Movimientos ---------- 54
4.1.2 Lista Tabu ---------- 55
4.1.3 procedimientos Básicos Tabu Search ------- 55
4.1.4 Estrategia de intensificación y Diversificación56
4.2 Path Relinking ---------- 57
4.3.1 Procedimiento Path Relinking---------- 58
4.3 Generación de Columnas---------- 70
4.3.1 Algoritmo de generación de columnas----- 74
4.3.2 Método de generación de columnas ------- 77
5. **CONCLUSIÓN ---------- 81**
6. BIBLIOGRAFIA---------- 82

CAPITULO I: INTRODUCCION

1 Descripción del problema

En muchas aplicaciones de nuestra vida relacionadas con los procesos industriales surgen problemas que la Investigación Operativa engloba bajo el título de Problemas de Corte y Empaquetamiento.Estos problemas, desde el punto de vista de la complejidad, siguen atrayendo el interés de los investigadores que continúan desarrollando algoritmos exactos y heurísticos cada vez con mayor eficiencia; la configuración de las piezas constituye el patrón en el tablero en el que se cortan.

El problema de corte en dos dimensiones (TDC) consiste en cortar con máximo beneficio un tablero rectangular en un conjunto finito de pequeñas piezas rectangulares, Este problema tiene un extenso campo de aplicaciones en la industria y comercio. Aparece en el corte de madera, cartón, cristal, plástico, laminas metálicas, etc. El problema (TDC) se puede considerar como un subproblema o versión sencilla de un problema de corte más general en el cual se debe satisfacer toda la demanda de piezas a partir de un conjunto de tableros de distintos tamaños.

Una característica importante del problema de corte en su dimensionalidad presentó la siguiente clasificación:

1. Un problema unidimensional es aquel donde solamente una dimensión de la entidad almacenada es significativa la otra dimensión no son relativamente en la solución esta situación ocurre por ejemplo en el corte de la barra de acero y aluminio. El problema de corte unidimensional es un problema de optimización en el que se busca la forma más eficiente de cortar una barra o pieza de material de longitud conocida en piezas más pequeñas de diferentes longitudes.

El objetivo es minimizar los costos de producción al minimizar la cantidad de material desperdiciado. El problema se plantea de la siguiente si tiene una barra de longitud L y se deben producir n piezas de longitudes , cada corte realizado en la barra tiene un costo fijo, independientemente de la longitud de la pieza resultante. El objetivo es encontrar una secuencia de cortes que minimice el costo total de producción.

Existen diferentes algoritmos para resolver este problema, desde soluciones heurísticas hasta algoritmos exactos. Uno de los algoritmos exactos más conocidos es el algoritmo de corte óptimo (OPT), que utiliza programación dinámica para encontrar la solución óptima en tiempo polinómico.

2. Un caso 2-dimensional Es un tablero rectangular y en el proceso de ambas dimensiones son fijas irrelevantes en la solución un ejemplo es cuando se corta lámina de acero cristal o panel de madera Johnson et al (1997).

Morabito y García (1998)También podemos decir si se tienen varias piezas de material de dos dimensiones, cada una con un ancho y largo específicos, y se deben producir varias piezas más pequeñas de diferentes tamaños a partir de planchas de material de dos dimensiones de tamaño fijo. Cada corte realizado en la plancha tiene un costo fijo, independientemente del tamaño de la pieza resultante. El objetivo es encontrar una secuencia de cortes que minimice el costo total de producción.

Caso Tridimensional, llamado también problema de embalaje y empaquetamiento aparece en la carta de caja dentro de contenedores o sobre plataformas. En algunos casos este problema puede ser visto como un caso especial del problema de corte en dos dimensiones. (Gilmore y Gomory) (1965)Rijckaert (1981)Dowsland (1985) problemas de corte y empaquetamiento están estrechamente relacionadas debido a la dualidad entre el material y el espacio ocupado por este (Dyckhoff) (1988) Dyckhoft Desarrollo del clima de clasificación que generaliza la clasificación presentada por Hinxman (1980)

Este método permite identificar las propiedades más comunes de los problemas de corte según las siguientes características.

1. Tipo de asignación
(B) todos los tableros y una parte de las piezas demandadas

(V) Una parte de los tableros y todas las piezas demandadas

2. Surtido de tablero almacenados

(O) Un tablero

(I) Tableros idénticos

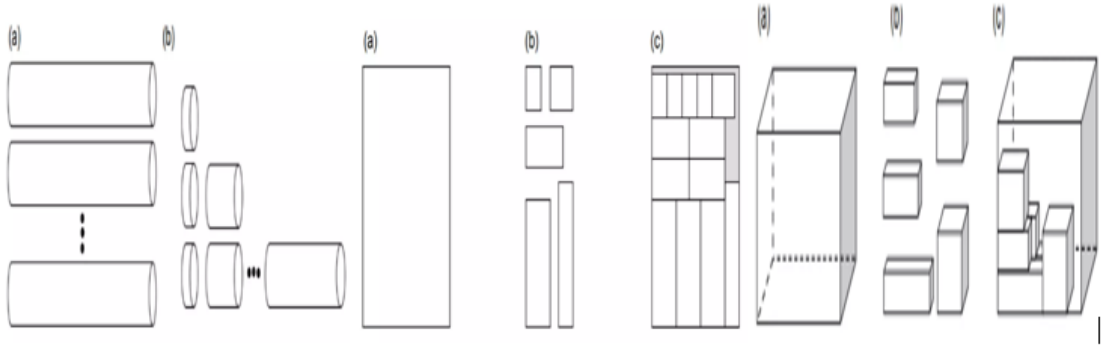
(V) Tablero diferente

3. Surtido de piezas demandadas

(F) pocas piezas de diferentes tamaños

(M) Muchas piezas de muchos tamaños

(R) Muchas piezas de relativamente pocas dimensionadas



De esta manera todo problema de corte se presenta según la tipología de Dyckhoff por $\alpha/\beta/\gamma/\delta$ así, por ejemplo, un problema tipo $(1/V/I/R)$ es un problema de corte en una dimensión donde se cortan todas las piezas demandadas que son muchas y de relativamente poco tamaño a partir de un solo tipo de tablero.

En este trabajo estudiamos los siguientes problemas de corte en dos dimensiones:

1. Problema de corte de tipo $(2/B/O/M)$de un único tablero se ha cortado la cantidad máxima posible de piezas demandada.

2. Problema de corte tipo $(2/V/V/M)$se debe cortar toda la demanda de piezas de diferente tamaño a partir de diferentes tableros rectangulares.

El problema de corte en dos dimensiones es más complejo de resolver que el problema de corte en una dimensión debido a la complejidad en determinar posibles patrones de corte realmente lo fundamental del problema en dos dimensiones es el proceso de generar los patrones de corte más que en el problema de corte en sí mismo como veremos más adelante para resolver el problema general $(2/V/V/M)$ utilizaremos como su problema el problema $(2/B/O/M)$.

En la literatura podemos encontrar un gran número de problemas de optimización qué se basan esencialmente en estructura lógica de los patrones de corte y empaquetamiento que incluyen: Cutting stock y Trim Lose Template Layout Coil Slitting, placement Packing y Depletion, Bin PackingStrip Packing, Vector Packing, Knapsack problema, Memory allocation y Multiprocessor Shedulling problemas.

1 JUSTIFICACION

La problemática del proceso Industrial ha obligado a buscar soluciones mediante la investigación operativa al estudiar problema de corte en dos dimensiones con el objetivo de obtener un Máximo beneficio a partir de un conjunto de pequeñas piezas rectangular a partir de una o múltiples tableros rectangulares de diferentes tamaños, el beneficio puede ser maximizar el valor total de las piezas cortadas minimizar el costo del material usada o minimizar el desperdicio de material.

El algoritmo Constructivo se basa en el cálculo de cotas superiores sobre las piezas. El algoritmo GRASP utiliza el algoritmo Constructivo para construir una solución y en la fase de mejora considera la fusión de rectángulos desperdicio con sus piezas adyacentes para cortarlos nuevamente con el algoritmo Constructivo posiblemente con mayor valor. Con respecto al algoritmo Tabú Search se consideran los siguientes elementos. Se realizan movimientos que mantienen la propiedad de corte guillotina.
La selección del movimiento considera un rectángulo al azar y todas sus piezas adyacentes y luego selecciona como movimiento el de mejor función objetivo.
El problema de corte en dos dimensiones consiste en determinar un conjunto de patrones variables, de tal forma que satisfagan los requerimientos de piezas de material solicitados, utilizando el menor número de láminas disponibles.

Los cortes en dos dimensiones se clasifican por el tipo de herramienta a utilizar como el corte por guillotina y corte no guillotina, considerados cortes rectangulares aplicados en el sector de papel, vidrio, metal, madera. (Pearl, 1984).

La aplicación de este tipo de problemas es el estudio de la problemática concreta en el corte de vigas estructurales, en una empresa de transformados metalúrgicos. El principal objetivo de este problema es reducir la cantidad de desperdicio de material que se origina al momento de realizar determinado corte.

En este sentido, esto permite construir un algoritmo constructivo, para lo cual se propone la utilización de un algoritmo metaheurístico GRASP (Greedy), el cual se utiliza para dar solución al problema de corte en dos dimensiones.

En los últimos años el desarrollo de procedimientos que utilizan heurísticas para resolver problemas de este tipo, han sido enormes. Ruiz Rivera Ruiz Lizama, Algoritmo GRASP para cortes de guillotina, 2006.

Este problema consiste en satisfacer las demandas de unas determinadas piezas, que se deben obtener mediante el corte o partición de un objeto más grande. La asignación generalmente se realiza persiguiendo el objetivo de minimizar los recortes o piezas residuales generadas por el corte y/o conseguir

la máxima utilidad. El algoritmo heurístico utilizado en esta aplicación informática es el algoritmo constructivo o algoritmo Greedy propuesto por Martelo y Toth en 1990.
La justificación de este problema radica en la necesidad de maximizar la eficiencia en el uso de materiales y minimizar los desperdicios. Al encontrar la mejor manera de cortar las planchas o láminas, se pueden reducir los costos asociados a la adquisición de material, disminuir los residuos generados y optimizar los procesos de producción.

Además, el problema de corte en dos dimensiones también tiene implicaciones en la planificación y la logística. Al optimizar el corte de las piezas, se puede mejorar la capacidad de almacenamiento y transporte, reducir los tiempos de producción y cumplir con los plazos de entrega de manera más eficiente.
Desde una perspectiva económica, el corte eficiente de materiales en dos dimensiones tiene un impacto directo en la rentabilidad de las empresas. Al minimizar los desperdicios de material y maximizar la utilización de recursos, se reducen los costos de producción y se mejora la competitividad en el mercado.
El ahorro en materiales puede ser especialmente relevante en industrias donde los costos de materia prima son altos, como la industria maderera o metalúrgica.

Además, el problema de corte en dos dimensiones también puede tener implicaciones en la planificación de la producción y la logística. Una optimización eficiente del corte permite una mejor organización del flujo de trabajo, reduciendo

los tiempos de producción y mejorando la utilización de los recursos disponibles. Esto se traduce en una mayor capacidad de respuesta a las demandas del mercado, plazos de entrega más cortos y una mejora general en la eficiencia operativa.

En términos ambientales, el problema de corte en dos dimensiones puede contribuir a la reducción de residuos y al uso más eficiente de los recursos naturales. Al minimizar los desperdicios de material, se disminuye la cantidad de desechos que van a parar a los vertederos, lo que a su vez reduce el impacto ambiental asociado a su disposición final. Además, una mayor eficiencia en el uso de recursos puede ayudar a conservar y proteger el medio ambiente.

2 OBJETIVOS

1. Revisar el estado del arte sobre problema de corte en dos dimensiones para comprender los últimos aportes a este problema, utilizando revistas científicas y tesis de postgrado del área de conocimiento.

2. Diseñar Metodológicamente algoritmos heurísticos para el problema de corte en dos dimensiones (TDC) mediante procedimientos constructivos.

3. Diseñar Metodológicamente algoritmos Metaheurísticos para problemas de corte en dos dimensiones (TDC) mediante procedimientos de Búsqueda Tabu, Path Relinking y Generación de columnas.

CAPITULO II: REVISION DEL ESTADO DEL ARTE

El problema de corte tiene un extenso campo de aplicación en la industria y comercio, especialmente en economía a gran escala donde algunos tipos de objetos se producen en grandes cantidades para satisfacer la demanda de los clientes.

Aparecen el corte de lámina metálica para automóviles, en el corte de paneles de madera para hacer muebles, en el corte de cartón para hacer caja, también en la producción de lámina de cristal, papel, plástico, acero etc.
Algunas aplicaciones del problema de corte en una dimensión son los trabajos de Wascher y Muller (1936) [117] en el corte de barra de acero y Gilmore y Gomory (1963) [57] en el corte de bobina de papel.

Aplicaciones de dos dimensiones son los trabajos de vasco et. Al (1991) [112] , Gemmill (1992) [54] y Agrawal (1993) [2] que resolvieron problemas de corte en la industria de acero. Farley (1990) [42] resolvió un problema en la industria textil Dyson y Gregory (1974) [37] tratarán problemas de corte en la industria de cristal.
Por otra parte, yanasse et. al (1991) [118] y Morabito y Garcia (1998) [91] presentaron trabajo sobre problemas de corte en la industria de madera. Matemáticamente él estudia el problema de corte se puede incluir dentro de la teoría de optimización combinatoria dado que tenemos una función objetivo a minimizar y un conjunto de soluciones finito.

Esta solución se conoce como patrones de corte y se representan las posibles combinaciones de las piezas sobre el tablero. Este problema admite una formulación matemática en términos de un modelo de programación lineal entera la primera formulación del problema de corte fue dado a conocer en el año 1939 por el premio Nobel de economía, L. Kantarovich (1960) [79].

Los primeros y más significativos avances en resolver problema de corte han sido los trabajos (1961) [1965] de P. Gilmore y R. El problema de corte se considera un problema es extremadamente difícil de resolver exactamente ya que la mayor parte de los problemas que modelizan situaciones de interés real suelen ser de gran tamaño cuando el problema de corte se expresa como un problema de programación lineal entera se tiene un número demasiado grande de variables patrones de corte lo que imposibilita en la práctica su resolución.

En los últimos 60 años el problema de corte en dos dimensiones ha sido extensamente tratado en la literatura de la investigación operativa utilizando los avances técnicos de cada época especialmente la tecnología computacional.

Gran parte de las investigaciones se han centrado en encontrar herramientas fundamentales para el planteamiento y resolución de problema de grandes dimensiones tales como el algoritmo heurístico que proporciona rápidamente una solución cercana a la solución óptima y algoritmo exacto que aun no siendo polinómico están siendo paulatinamente más eficiente.

Sobre el problema de corte ha sido publicado por varios trabajos que resumen la mayor parte de los métodos que se han propuesto para resolverlo algunos de estos son Surveys son trabajo de Golden (1976) [62], Madsen (1980) [83] Hinxman (1980) [76], Dyckhoff (1988 − 1990) [32] [33], Dyckhoff y Finke (1992) [35] .

Por otra parte, Sweeney y Paternoster(1992) [104] identificaron más de 400 publicaciones que trabajaba con aplicaciones relativa a problemas de corte y empaquetamiento y más recientemente. El maghraby et al. (2000) [38]hacen referencia a más de 800 publicaciones que tratan sobre los mismos problemas.

Alguno de los métodos que se han utilizado para intentar de resolver problemas de corte son los siguientes:

a. método de programación lineal: Gilmore y Gomory (1965) [58],Israni y Sanders (1982) [77], wascher(1990) [116], Vanderbeck(1999) [107],Alvarez Valdes et al.(2001) [4]

b. método de Programación dinámica: Christofides Whitlock (1977) [20] Beasley (1985) [9], Scheithauer y Terno(1988) [101], Gochet Vandebroek (1989) [61], Christofides Hadjiconstantinou (1995) [17]

c. Algoritmos Branch and Bound y Branch and Price. Viswanathan

Bagechi(1993) [114], Vanderbeck(2000) [108].

d. procedimiento inteligencia basado en sistema expertos: Dagli(1990) [23], Elmaghraby(2000) [108]

e. Método basado en procedimiento heurístico que incluyen estrategias de tipo Grasp (Greedy Randomized Adaptive Search Procedure) y Tabu Search. Álvarez Valdés et al.(2000) [3]

f. Problema Knapsack en una dimensión Fayard et al. (1998) [46]

g. Simulated Annealing (AS) parada et al. (1998) [94], Hildegard y Wascher (1998) [74]

El extenso número de publicaciones en esta área no es sorprendente debido a las siguientes situaciones:

1. El problema de corte en dos dimensiones está clasificado como $Np-duro$ (Garey y Johnson (1979) [53]por lo que se han propuesto diferentes métodos heurísticos para su resolución.

2. No hay un método común para resolver las diferentes clases de problemas de corte cualquier cambio en la configuración del problema afectará la eficiencia de la solución empleada.

3 Restricciones

Generalmente en el proceso para obtener las piezas se imponen muchas restricciones sobre la forma en que se puede cortar el material. Algunas restricciones más comunes que usualmente se consideran sobre patrones de corte son las siguiente:

2.1.1 Cortes Ortogonales

Una restricción especifica que un patrón de corte es ortogonal, si todos los cortes se hacen paralelo a lados del tablero rectangular. En este caso ha sido considerado por numerosos autores, por Gilmore y Gomory (1998) [91] . Por otra parte, casos de patrones no ortogonal surge cuando se corta una pieza que tiene alguna de las dimensiones más grande que el largo y ancho del tablero.

Los cortes ortogonales se refieren a secciones rectas que se realizan de manera perpendicular a los ejes cartesianos (horizontal y vertical). Estos cortes se utilizan para dividir o seccionar el tablero en partes más pequeñas y analizar o representar información específica. Imagina un tablero de ajedrez o un tablero cuadriculado con filas y columnas.

Si realizamos un corte ortogonal horizontal, dibujamos una línea recta que atraviesa el tablero en dirección horizontal, perpendicular a las filas. Esto divide el tablero en dos partes: la parte superior y la parte inferior.

De manera similar, si realizamos un corte ortogonal vertical, dibujamos una línea recta que atraviesa el tablero en dirección vertical, perpendicular a las columnas. Esto divide el tablero en dos partes: la parte izquierda y la parte derecha.

Estos cortes ortogonales en un tablero en dos dimensiones se utilizan en diversos contextos, como juegos de mesa, programación gráfica, representación de datos y análisis espacial. Por ejemplo, en un juego de ajedrez, los cortes ortogonales se utilizan para delimitar las áreas de juego y distinguir las diferentes posiciones de las piezas. En programación gráfica, se pueden usar para crear efectos visuales o realizar transformaciones en objetos en el tablero. Los cortes ortogonales en un tablero en dos dimensiones implican realizar secciones rectas perpendiculares a los ejes cartesianos. Estos cortes se utilizan para dividir el tablero en partes más pequeñas y analizar o representar información específica en diferentes contextos.

2.1.2 Corte Guillotina

Una característica frecuente en los procesos industriales es que los cortes de sierra se hacen interrumpidamente de un lado del tablero al lado opuesto y paralelo a los dos lados sobrante. Esta clase de corte se conocen como corte de tipo guillotina. Algunos autores que han estudiado este problema son Gilmore y gomory $(1965 - 1966)$ [58] .

El corte guillotina se puede referirse a una operación de corte que se realiza en línea recta y de manera vertical en el tablero, similar al movimiento de una guillotina. Este tipo de corte se utiliza para dividir o seccionar el tablero en partes más pequeñas.

Imaginemos un tablero cuadriculado con filas y columnas. Si realizamos un corte guillotina vertical, trazamos una línea recta vertical desde la parte superior hasta la parte inferior del tablero, dividiéndolo en dos secciones. Este tipo de corte se asemejaría al movimiento de una guillotina descendiendo verticalmente.

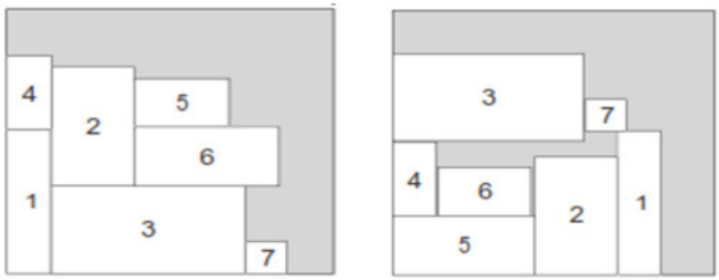
Del mismo modo, si realizamos un corte guillotina horizontal, trazamos una línea recta horizontal desde el extremo izquierdo hasta el extremo derecho del tablero, dividiéndolo en dos secciones. Nuevamente, este tipo de corte sería similar al movimiento de una guillotina descendiendo de manera horizontal.

El cortes guillotina en un tablero en dos dimensiones se utilizan en diversas aplicaciones, como juegos de mesa, diseño gráfico, programación y trabajos de corte y división. En los juegos de mesa, los cortes guillotina pueden utilizarse para separar áreas de juego o dividir el tablero en secciones para diferentes propósitos.

2.1.3 Corte en etapas

Esta restricción limita a que los corte tipo guillotina se deben hacer en un número de distintas "Etapas". En la primera etapa los cortes son paralelos a uno de los lados del rectángulo y en las etapas siguiente los cortes son ortogonales.

Al corte realizado anteriormente, y así se sigue cortando sucesivamente hasta terminar el proceso de corte sobre el tablero. Si existe una cota superior sobre el número de etapas a realizar, entonces se obtienes patrones de corte en n- etapas.



Corte Guillotina Corte no Guillotina

El problema de corte en dos dimensiones, consiste en cortar de un tablero rectangular de dimensiones fijas un conjunto finito de pequeñas piezas rectangulares con máximo beneficio. El problema TDC se puede considerar como un subproblema de problema de corte más general en el cual se deben satisfacer un conjunto de diferentes pedidos de piezas cortándolas a partir de un conjunto de tablero de distintos tamaños.

Una instancia del problema TDC viene dada por cuádruple (R, S, v, d) donde $R = (L, W)$ es un tablero rectangular de longitud L y anchura W, el conjunto $S = \{(l_1, w_1)\,(l_2, w_2)\,....\,(l_m, w_m)\}$ contiene las piezas de longitud li y anchura wi Cada pieza i tiene un valor vi y una demanda di. El problema es cortar el rectángulo R en xi copias de cada pieza i talque $0 \leq x_i \leq d_i,\ i = 1, \ldots m$ y la utilidad total $\sum vixi$ sea máxima.

Un patrón de corte es una secuencia de corte en R de piezas rectangulares, las piezas producidas que no se incluyen en S son el desperdicio. En este trabajo se imponen a los productos de cortes dos restricciones adicionales que usualmente aparecen en la aplicación de la vida real.

1. Las piezas tienen orientaciones fijas, esta restricción aparece cuando el material a cortar tiene algunas veta o dibujo y por tanto una pieza (l, w) es diferente de otra pieza (w, l) en el patrón de corte Si se permite la rotación de las piezas, entonces los algoritmos que hemos desarrollado se pueden modificar fácilmente para incorporar esta variante.

2. Solamente se permitirá corte guillotina, esta relación aparece frecuente en la mayoría de los procesos de cortes industriales. esto se debe a que la tecnología involucrada solamente permite hacer cortes de un lado del rectángulo al lado opuesto y paralelo a los dos lados sobrantes. Por su parte el problema de corte no guillotina es completamente diferente y tiene que ser resuelto usando otros métodos. (ver, por ejemplo, Beasley (1985)[10],chauny y Loulou (1994)[16].

Otra restricción que aparece algunas veces es las limitaciones en el número de etapas en la cual se realizan los cortes. Sin embargo, en este trabajo no consideramos esta restricción y se permiten patrones de corte sin límite de etapas. Siguiendo la clasificación propuesta por Fayard et al (1998)[46] para el problema TDC, se pueden distinguir cuatro versiones.

a. La versión no restringida sin peso ($UUTDC$) una instancia se define por la cuádruple $(R; S; s, \infty)$. El valor u_i de cada pieza i es igual a la superficie $S_i = l_i w_i$. La función objetivo a maximizar es la superficie total ocupada lo que equivale a minimizar el desperdicio.

b. La versión no restringida con peso($UWTDC$) la instancia se presenta por $(R; S, v, \infty)$ donde los valores de las piezas son independientes de superficie y el objetivo es maximizar el valor total de las piezas cortadas.

c. La versión restringida sin peso ($CUTDC$) una instancia se describe por (R, S, s, d), donde cada di limita el número de piezas de tipo i a cortar y por lo menos algún d_i es estrictamente menor que $\left[\frac{L}{l_i}\right]\left[\frac{W}{w_i}\right]$ que es el número máximo de piezas i que pueden obtenerse de R.

d. La versión restringida con peso($CWTDC$) las instancias se definen por (R, S, u, d) y es el caso más general.

2.2 Método de solución

En esta sección hacemos una descripción general sobre algunos de los métodos existentes en la literatura con la cuales se ha intentado resolver el problema de corte guillotina en dos dimensiones. No pretendemos hacer una exposición exhaustiva de los mismo, sino solamente presentar una versión panorámica de las aportaciones que no parecen más significativa.

2.2.1 Versión no restringida del problema TDC

El problema de corte de la guillotina no restringida en dos dimensiones ha sido considerado por pocos autores de la literatura. Algunos de los trabajos más importante sobre este problema son los siguiente.

Método Exactos

2.2.2. Beasley (1985)[9]

En 1985, J.E Beasley presento una correcta recursión en programación dinámica que supera el error encontrado en la recursión dada por Gilmore y Gomory (1966)[59] . Para el problema de corte de guillotina no restringida en n etapa Beasley extendió este resultado al problema de corte de guillotina general (sin límite el número de etapas) y desarrollo un método heurístico que acelera el procedimiento cuando la recursión llega ser computacionalmente imposible.

Problema de corte en n etapas sean $A_o = (L_o, W_o)$ el tablero rectangular inicial (l_i, w_i), $i = 1 \ldots m$ las dimensiones de la pieza demandada $L = \{1.2 \ldots . Lo - 1\}$ y $W = \{1, 2, \ldots . W_o - 1\}$ los conjuntos posibles de cortes paralelos a los ejes x e y, y sean $F(K, x, y)$ y $G(K, x, y)$ los valores de un patrón de corte son paralelos al eje x o paralelo a y respectivamente. Se pueden identificar las siguientes situaciones de desperdicios en (x, y) .

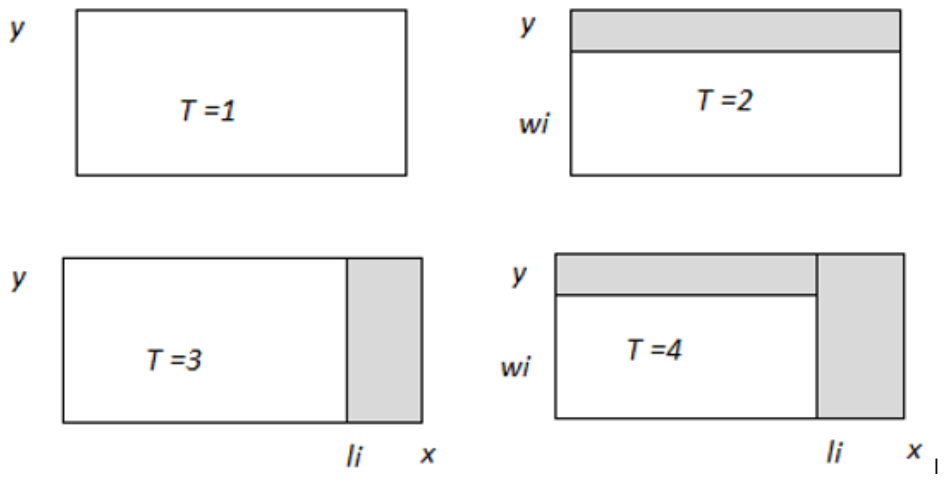
$$max = \{0, v_i | \, l_i = x, w_i = y, \ i = 1 \ldots m\} \ T = 1$$

$$\mathrm{F}(0, x, y) = \quad max = \{0, v_i | \, l_i = x, w_i \leq y, \ i = 1...m\}, \ T = 2$$

$$max = \ \{0, v_i | \, l_i \leq \ x, w_i = y, \ i = 1...m\}, T = 3$$

$$max = \{0, v_i | \, l_i \leq \ x, w_i \leq y, \ i = 1...m\}, T = 4$$

$F\,(\ n, \ L_o, W_o)\, y\, G\,(n, L_o, W_o)$ son los valores de un patrón de corte óptimo en etapas sobre el rectángulo A_o dependen de la dirección del primer corte y se pueden obtener a partir de las siguientes ecuaciones recursivas básica:



$$F\,(K, x, y) = max\,\{F\,(k, x, y) + (K, x, y - y_1)\, y_1 \epsilon W\}$$

$$y_1 \epsilon \ k_1\,(y)\,; \ G\,(k-1, x, y)$$

$$\mathrm{G}(K, x, y) = max\,\{o, x, y; \ G\,(k, x_1, y) + G\,(k, x - x_1, y)\,; x_1 \epsilon L, \ x_1 \leq k_2\,(x)\,; \ F\,(k-1, x, y)\}$$

Donde

$k_1(y) = \left\{ \frac{y}{2} \; si \; T \; = 2 \; o \; T \; \acute{=} 4 \; \acute{o} \; y - 1 \right\}$ en otro caso

$k_2(y) = \left\{ \frac{x}{2} \; si \; T \; = 3 \; o \; T \; \acute{=} 4 \; \acute{o} \; x - 1 \right\}$ en otro caso

Introduciendo patrones normalizado se redefinen $l \; y \; W$ tal que:

$$L \; = \left\{ x | x = \sum_{i=1}^{ni} l_i a_i \, , 1 \leq x \leq L_o - k_3, a_i \geq 0 \; y \; entonces, i \; = 1 \ldots m \right\}$$

$$W \; = \left\{ y | y = \sum_{i=1}^{m} w_i b_i \, , 1 \leq y \leq W - k_4, b_i \geq 0 \; y \; entonces, i \; = 1 \ldots m \right\}$$

Donde

$k3 = \{\min(li \,|i = 1 \ldots m) \; si \; T = 3, \; T = 4 \; ; 1 \; en \; otro \; caso\}$

$k4 = \{\min(wi \,|i = 1 \ldots m). \; si \; T = 2; \; T = 4 \; o \; 1 \; en \; otro \; caso\}$

En la cual tenemos las ecuaciones básicas:

$p(x) = max \; \{0, x \,| x_1 \leq \; x, \; x_1 \epsilon L, \; x < L_0\}$

$q(x) = max \; \{0, y_1 \,| y_1 \leq y, \; y_1 \epsilon W, \; x < W_0\}$

Por lo tanto, no todos los valores de (x, y) son necesarios y con $p(x)$ y $q(x)$ se obtiene:

$F(K, x, y) = max \{F(0, x, y) \; ; \; F(k, x, y_1) + F(k, x, q(y - y_1) \; ; y_1 \epsilon W, \; y_1 \leq k_1(y) \; ; G(k - 1, x, y)\}$

$k \geq 1, x \epsilon L^0, \; y \; \epsilon W^o$

$$G(K,x,y) = max\{G(0,x,y);\ G(k,x_1,y) + G(k,p(x-x_1),y); x_1\epsilon\ L,\ x_1 \leq k_2(x); G(k-1,x,y)\}$$

Donde $L^0 = L \bigcup \{L_0\}$ y $W^0 = W \bigcup [w_0]$

La ecuación recursiva modificada es computacionalmente más efectiva qué las ecuaciones recursivas básicas Ya que involucran solamente operaciones que requieren en una memoria de números. Sin embargo, las ecuaciones recursivas mejorada también llegan a ser prácticamente imposible sí $|L|\ o\ |W|$son demasiado grandes. en tal caso, es posible Modificar el algoritmo óptimo dado anteriormente y obtener un procedimiento heurístico posible computacionalmente. Corte general (sin límite en el número de tapas) Beasley adapto la recursión en etapa al problema general, considerando ilimitado el número de etapas.
Sea $F(0,x,y)$ han definido para $T = 4yF(-,x,y)$ el valor óptimo de un patrón de corte guillotina general sobre un rectángulo(x,y).
Entonces:
$F(-x,y) = max\{F(0,x,y);\ F(-x,y_1) + F|-x,q(y-y_1)|,\ y_1\epsilon\ W,\}$
$y_1 \leq \frac{y}{2}; F(-,x_1,y) + F[-,p(x-x_1),y], x_1\epsilon\ L, x_1 \leq x/2$
$x_1\epsilon L^0,\ x_1\epsilon W^0$
Debido a que la recursión del problema general también llega a ser computacional imposible, se introduce el siguiente procedimiento heurístico que redefine$L\ y\ W$ para asegurar$|L|\ y\ |W|$ que son pequeños. Sea M una Cota sobre $L\ tal\ que\ |L| \leq M$.

1. Sea $N = \{1,2\ldots.m\}$ el numero de piezas a cortar de A_o
2. Calcular

$$L = \left\{ x \left| x = \sum_{i \epsilon N}^{i} l_i a_i \ , \ 1 \leq x \leq L_0 - min\left[l_j \left| j \epsilon N \right. \right], a_i \geq 0 \ y \ enteras, \ i \epsilon N \right. \right\}$$

3. Si $|L| \leq M$ entonces para con el conjunto requerido L; en caso contrario ir a etapa 4.

4. Definir $L_j = min\left\{ l_i \left| i \epsilon N \right. \right\}$, hacer $N \ = N - \{j\}$ e ir a la etapa 2.

Con $l \ y \ W$ redefinido se obtiene la siguiente recursión heurística en programación dinámica. W se halla de forma similar. Con L y W redefinidos se obtiene la siguiente recursion en programacion dinamica:

$F\left(-x, y\right) = max\left\{ F\left(0, x, y\right); \ F\left(-x, y_1\right) + F\left| -x, q\left(y - y_1\right) \right| \right\}$

$y_1 \varepsilon \ W, \ y_1 \leq y - 1; \ F\left(-x_1, y\right) + F\left| \left(-, p\left(x - x_1\right), y\right) \right|,$

$x_1 \epsilon \ L, \ x_1 \leq x - 1 \}, \ x \epsilon L^0, \ y \epsilon W^0$

Donde

$$F\left(0, x, y\right) = \ max\left\{ 0, \left\lceil \frac{x}{L_i} \right\rceil \left\lceil \frac{y}{W_i} \right\rceil v_i \ , L_i \leq x, \ w_i \leq y, i = 1....m \right\}$$

El algoritmo desarrollado por el autor permite resolver exactamente instancia de problemas de corte guillotina no restringido de tamaño pequeño y medianos y de obtener buena solución aproximada para problema generales.

2.2.3. Hifi y Zissimopoulos (1996)|71|

En 1996. M. Hifi y V. Zissmopoulos Desarrollar un algoritmo que se basa en una propiedad recursiva que caracteriza a los problemas de corte guillotina no restringidos.

Esta propiedad permite considerar todas las secciones horizontales y verticales en cada sub rectángulo qué se produce en el proceso de corte y de esta forma hallar una solución óptima sobre el rectángulo inicial.
Los autores utilizan problema knapsack unidimensional resuelto en programación dinámica para obtener eficientes cotas inferiores y superiores, también se usa un importante criterio de optimalidad que permite reducir considerablemente la ramificación en el desarrollo del árbol de búsqueda.

El procedimiento resuelve ambas versiones del problema de corte no restringido. sin peso (UU_TDC y con peso UW_TDC y se considera una generalización de los algoritmos propuesto por Herz.(1972) |67| y Hifi y Zissmopoulos(1996) |72|.

2.3. Métodos Aproximados

2.3.1. Morabito et al. (1992) |90|

En 1992 R. Morabito, M. arenales y V. Arcaro estudiaron el problema de generación de patrón de corte para un tablero rectangular y desarrollaron un tipo de estructura denominada $AND/0R - GRAPH$ para presentar la solución del problema que se resuelven al descomponer los en un conjunto de problemas más pequeños te resuelven totalmente.
Este método se usa comúnmente en el campo de la Inteligencia artificial, para la búsqueda sobre el grafo los autores combinan la estrategia clásica.$depth - firt\ y\ hill-climbing$ para crear el algoritmo DH/HC definen el problema de corte como un problema de búsqueda sobre un grafo dirigido dirigido donde

el nodo inicial representa el rectángulo inicial y los nodos finales representan las piezas demandadas.

Un camino completo sobre el grafo representa una secuencia de corte a partir de un rectángulo inicial y que finaliza en una pieza demandada generando de esta forma un posible para patrón de corte. La presente una secuencia de corte guillotina horizontales y verticales sobre el rectángulo inicial $A_0 = (L_0, W_0)$ que produce el patrón de corte. Inicialmente A_0 se corta verticalmente produciendo dos rectángulos intermedios $B\ y\ C$ llamados Sucesores de A_0 después ambos rectángulos $B\ y\ C$ son cortados de forma independiente.

Un corte horizontal sobre B y otro corte horizontal sobre produce los Sucesores $D\ y\ E$ y finalmente F se cortan verticalmente produciendo los rectángulos son rectángulos intermedios y se omiten en el patrón de corte, el resultado obtenido indica que el algoritmo DF/HC parece ser una buena alternativa para resolver problemas de gran tamaño donde la memoria y el esfuerzo computacional son factores esenciales.

2.3.2 Fayard y Zissmopoulos (1995)[48]

En 1995 D. Fayard y V Zissmopoulos presentaron un algoritmo heurístico para resolver el problema en dos dimensiones TDK ($del\ ingles\ Two\ Dinensional$. El algoritmo ODK ($delingles, OneDimensionalKnapsack$) para generar un conjunto de franjas óptima y después la justan de forma óptima dentro del

rectángulo inicial usando otro problema ODK ($Martello\, y\, Toth$(1976) [84],para reducir el número de problema ODK involucrado los autores utilizan el método de programación dinámica y demuestran que en el procedimiento son eficientes solamente a problema knapsack.

El algoritmo propuesto llamado BSC ($delinglesBeststripscutting$) se deriva del análisis de la estructura de patrones de corte óptimo la idea básica del algoritmo BSCse resumen en los siguientes procedimientos:

1. Procedimiento de discretización DP ($de\ lingles\ Discretisation\ procedure$) esta etapa permite tratar con un número finito de sub rectángulo candidato a dar mejor solución, se utilizan los patrones de corte de guillotina normalizado donde cada corte de guillotina se hace solamente en algunos puntos qué son combinaciones lineales de las longitudes de las piezas consideradas en el Su rectángulo.

En particular para (α, β) estos puntos definen los siguientes conjuntos.

$PL = \left\{ x | \, x = \sum_{i=1}^{m} z_i l_i \leq \frac{1}{2} L,\ z_i \geq 0,\ y\ enteras \right\}$

$PH = \left\{ x | \, x = \sum_{i=1}^{m} z_i h_i \leq \frac{1}{2} H,\ z_i \geq 0,\ y\ enteras \right\}$

$P_L\, (y\ similarmente\ P_H)$ puede ser generado considerando la función:

$0, si\ i = 0\ x = 0\ \infty, si\ i = 0\ x0 \qquad g_i - 1\,(x)\,,\ si\ x < l_i\ \wedge\ i \epsilon\ I$

$min\left\{ g_i - 1\,(x)\,,\ max\left\{ 1,\ min_k \left\lceil g_i - 1\,(x - k l_i)\,, 1 \leq k \left[\frac{x}{l_i} \right] k\ \epsilon\ N \right\rceil \right\} \right\},\ i \epsilon\ I,\ x \epsilon\ X$

Donde: $I = \{1, ...m\}\ y\ X = \left\{ 0, 1, 2, 3, ... \left[\frac{1}{2} L \right] \right\}$ por lo tanto, tenemos un punto

$x\ \epsilon P_l\ si\ g_m\,(x) = 1\ y\ x \notin P_L\ si\ g_m\,(x) = \infty$

2. Procedimiento de generación SEP (*del ingles Strips generation procedure*) para crear las franjas horizontales se reordenan las piezas a cortar en el conjunto $Sk_1 \subset Sk_2 \subset \ldots \subset Sk_n$ para cada Sk_i se asocia un problema unidimensional definido de la siguiente manera.

$(ODK\,(i))$

$max\ \sum_{l_\mu \epsilon Sk_\mu} \pi_\mu x_\mu$

$sujeto\ \sum_{l_\mu \epsilon Sk_\mu} \pi_\mu x_\mu \leq \alpha, x_\mu \geq 0,\ x_\mu\ enteras$

Donde α es la longitud del tablero a cortar π_μ es el beneficio asociado con la pieza (l_μ, h_μ) y x_μ $eselnúmerodevecesquelapieza(l_\mu, h_\mu)$ aparece en la $i-esima$ franja resolviendo r problemas $ODK\,(i)$ se crean r franjas horizontales con longitud α altura h_{ki} y valor $F_i\,(\alpha)$,esta franja horizontales son óptimas con respecto de $\alpha\ y\ h_{ki}$ las (franjas verticales si en el mismo procedimiento.)

3. Procedimiento de llenado de rectángulo $FP\,(del\ ingles\ Filling\ procedure)$. En esta etapa se selecciona la mejor es franja para rellenar el rectángulo $\alpha, \beta)$ para corte horizontal es el problema knapsack respectivo se define de la siguiente manera.

$$(ODK\,(R+1))$$

$$F_{r+1}\,(\beta) = max \sum_{i=1} F_i\,(\alpha) y_i$$

$$Sujeto \sum_{i=1} \beta_i y_i \leq \beta,\ y_i \leq 0,\ y_i\ enteras$$

Dónde β es la altura del rectángulo a cortar $y_i = 1 \ldots r$ denota el número de repeticiones de la franja en el caso de las franjas verticales se define el problema $ODK\,(v+1)$ y se sigue el mismo procedimiento.

Por lo tanto,los patrones de corte generados tienen valores.

$$V \ = F_{r+1}\,(\beta) \ \ y \ V = F_{r+1}\,(\alpha)$$

Un procedimiento BSC propuesto por Fayard y Zissmopoulos trata eficientemente con el problema de gran tamaño qué son difícil de manejar por otros métodos conocidos, cuales con frecuencia están limitados por facilidades de memoria el alcoholismo puede ser generalizado por tratar con otras versiones del problema TDK.

2.3.3 Hifi (1997) ⌈68⌉

En 1997, M, Hifi desarrollo un método híbrido que utiliza programación es dinámica y método de Inteligencia artificial para resolver el problema de corte guillotina no restringido sin peso ($UUTDC$) y con peso ($UWTDC$),el procedimiento propuesto denominado combina eficientemente lo siguiente algoritmo heurístico de la literatura.

1. El algoritmo DH usa procedimiento de árbol de busca introduciendo la estrategia.$depth - first - seach\ y\ hill - climbing$ el algoritmo DH trata solamente en el caso sin peso ($Morabito\ et\ al.$) (1992) |90|).

2. El algoritmo KD es un procedimiento que se basa en una serie de problemas unidimensional. Usa técnicas de programación dinámica para dar una buena

solución sub óptima pero no explota los rectángulos internos. Fayard y Zissmopoulos (1995) [48].

El algoritmo DH/KD propuesto por $hifi$ se inicia con una buena Cota inferior que obtiene mediante el algoritmo KD y en cada nivel del árbol de búsqueda usa KD el algoritmo para obtener una nueva Cota inferior y utiliza el problema knapsack unidimensional para construir cada Cota Superior refinada. El método híbrido DH/KD resulta de ser eficiente para resolver problemas de gran tamaño, en la actualidad este algoritmo está considerado el más eficiente para problemas no restringido.

2.4. Versión restringida del problema TDC

Métodos Exactos

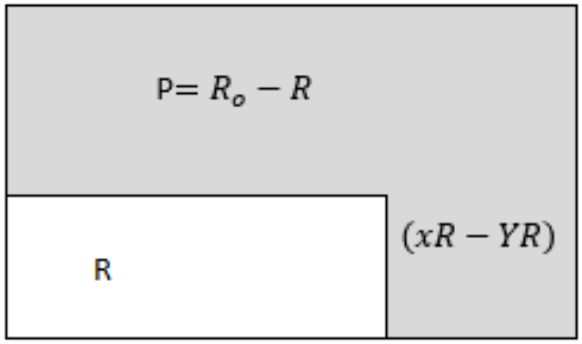
2.4.1 Christofides y Whitlock (1977) [20]

En 1977, christofides y whitlock propusieron un algoritmo *Branch and Bound* de búsqueda de profundidad ($depth - first$) para el problema de corte guillotina restringido. En este procedimiento todos los posibles patrones de corte son enumerados mediante la construcción de un árbol.

El nodo raíz del árbol representa el tablero inicial, los otros nodos del árbol representan la colección del rectángulo obtenido en el proceso de corte por medio de cortes de guillotina sobre el rectángulo inicial.

La ramificación sobre el árbol corresponde a corte de guillotina evitando duplicidades por simetría Y utilizando sólo Cortés normalizado. El algoritmo utiliza las funciones knapsack en dos dimensiones de *Gilmore y Gomory* (1966) [59].

Un método de evaluación basado en una rutina de transporte de $DesleryHakimi$ (1969) .para producir una Cota Superior sobre el valor máximo obtenido en cada nodo, Este procedimiento resulta adecuado para resolver problemas de corte con tamaño pequeño y mediano:



2.4.2. Viswanathan y Bagchi (1993)[114]

En 1993,$Viswanathan\ y\ Bagchi$ desarrollaron un método $Best - first - Search$ .Este método se basa en que todos los patrones de corte de guillotina se pueden obtener por medio de uniones verticales y horizontales de rectángulos introducida por $Wang$ (1983) [115] .
Si el rectángulo construido llamado rectángulo guillotina R no exceder las dimensiones del rectángulo inicial R_0 entonces el algoritmo pone R en la esquina inferior izquierda de R_0 $yexploralasuperficienoutilizada$P=$(R_0 - R)$usando la ecuaciones recursivas de $Gilmore\ y\ Gomory$ (1966) [59]

Para obtener una Cota superior sobre R_0. Sea $g\,(R)$el valor de la pieza contenida en $Ryh\,(R)$ el valor máximo obtenido, sin violar las restricciones

de demanda, de la región P, entonces $F(R) = g(R) + h(R)$ representa el valor máximo de un posible patrón de corte que contiene a $R, iswanathan\ y\ Bagchi$ Propusieron dos formas para calcular una estimación denota la dimensión de entonces un primer valor de Se obtiene por medio de la estimación.

$U_1(x_R, y_R) = F(L, W - y_R) + F(L - x_R, W)$

Dónde F es una función en dos dimensiones satisfaciendo que para toda

$x\ e\ y\ tal\ que\ x \leq L,\ y \leq\ W$

$F(x, y) = max\{F_0(x, y),\ F(x_1,\ y) + F(x_2,\ y);\ F(x, y_1)\, F(x, y_2)\}$

Donde $F_0(x, y) = max\{\{0, v_i\} :\ l_i\ \leq\ x, w_i \leq\ y,\ i = 1...n\ \}$

$x \geq x_1 + x_2,\ 1 \leq x_1 \leq x_2\ \ y\ y \geq y_1 + y_2,\ 1 \leq y_1 \leq y_2$

La segunda estimación denotada $U_2(x, y)$ por esa también la función F es da como sigue:

$U_2(x, y) = max\{h_1(x, y), h_2(x, y)\}$

Donde:

$h_1(x, y) = max\{U_2(x + u, y) + F(u, y) :\ 1 \leq u \leq L - x\}$

$h_2(x, y) = max\{U_2(x, y + u) + F(u, y) :\ 1 \leq u \leq W - y\}$

$U_2 = (L, W) = 0$

El método enumera todas las posibles formas del rectángulo $R = (x_R, y_R)$ se puede poner en esquina inferior izquierda de algunos patrones de corte guillotina. una vez que $g(R)$ es conocido y $\dot{h}(R)$ ha sido estimado usando U_1 *ò* U_2 se puede calcular $\acute{f}(R) = g(R) + \acute{h}(R)$ que es una Cota Superior del patrón de corte de guillotina sobre Ro qué contiene R.

$Viswanathan\ y\ Bagchi$ mostraron en su trabajo cómo el método $Best - First\ search$ que es ampliamente utilizado para resolver problemas en el

campo de la Inteligencia artificial también es una excelente herramienta para resolver algunos tipos de problemas de la investigación operativa, tal como el problema de corte guillotina en dos dimensiones.

2.4.3 Christofides y Hadjiconstantinou (1995)[17]

En 1995, N christofiles y E. presentaron un algoritmo de árbol de búsqueda para resolver el problema de corte de guillotina restringido en dos dimensiones que es una mejora del algoritmo desarrollado por christofiles y Whitlock (1977)[20] .

El método limita el tamaño del árbol de búsqueda por el uso de una Cota superior ajustada que se deriva de una relajación del espacio de estado de la formulación en programación dinámica del problema original. más detalles sobre esta metodología se puede hallar en christofiles et al (1981)[18] [19] .

El procedimiento en el proceso de corte del rectángulo en términos de un árbol. la ramificación a partir del nodo raíz se corresponde a todos los posibles cortes sobre A_0 y nodo al final de esta ramificación representa los rectángulos producidos por los correspondientes cortes sobre A_0 .Cada nodo n representa un estado del rectángulo después que se ha realizado un corte.
Para generar todos los posibles patrones de corte se incluye una variable artificial $O - corte$ que deja los rectángulos intactos no son candidatos para futuros Cortes.

2.4.4 Hfi (1997)[69]

En 1997 $M.\ Hifi$ propuso una mejora en el procedimiento de $Viswanathan$ (1993) introduciendo cotas unidimensionales en el algoritmo original. utilizó propiedades de la programación dinámica para obtener buenas cotas inferiores y Superiores los que llevan a un a una significativa reducción en el árbol de búsqueda. el autor introduce una nueva Cota inferior al inicio del algoritmo y una Cota superior complementaria que se usa en cada uno del árbol.

Hifi, considero un heurístico propuesto por $Zissimopoulos$ (1984) [119] para obtener una Cota inferior sobre el rectángulo inicial. el heurístico usa un problema para crear un conjunto de franjas horizontales y después las combina para obtener un posible para patrón de corte. Se muestra que todas las franjas horizontales y verticales óptima se pueden obtener con solamente dos problemas unidimensionales acotado que se resuelven usando propiedades de programación dinámica ($Martello\ y\ Toth$(1990) [85]. La versión mejorada del algoritmo de incrementa la eficiencia sobre la instancia del tamaño pequeño y grande.

2.4.5 Hifi y Zissmopoulos (1997)[73]

En 1997,$M.\ Hifi\ y\ V.\ Zissmopoulos$ desarrollaron una versión mejorada del procedimiento presentado por $Christofides\ y\ Whitlock$ (1977) [20] para resolver el problema de corte y de guillotina restringido en dos dimensiones.

La eficiencia del algoritmo modificado se basa en la forma incrementar de resolver el problema de transporte, es una nueva estrategia de ramificación que reduce considerablemente el número de llamadas a la rutina de transporte y en la introducción del problema acostados para calcular una nueva Cota inferior y Superior.

2.4.6 Cung et al. (2000) [22]

V. Cung, *M. hifi y L. Cun* Presentaron en el año 2000 una nueva versión del algoritmo mejorado de *Viswanathan y Bagchi* propuesto por *hifi* (1997) [69]. Los auctores desarrollaron una mejora en la Cota inferior inicial, una Cota Superior complementaria para los dos inferiores y desarrollado una nueva estrategia de búsqueda; estás mejoras aumentan significativamente la eficiencia del procedimiento.

La mejora de la Cota inferior reduce el espacio de búsqueda y se basa en redefinir los conjuntos de combinaciones lineales de longitud y ancho P_α *y* P_β introducido por *Christofiles y Whitlock* (1977) [20] .

La Cota Superior complementaria se basa en explotar en cada todo el valor de la demanda residual de cada pieza considerada. se utiliza la estrategia denominada $G-elemento$en vez de la estrategia $Best_First_search$ para explorar el nodo con mejor solución posible.

Además, proponen una nueva representación de la lista *Clist* que guarda los subproblemas explorados y que permite acelerar el procedimiento, los autores muestran mediante un estudio computacional que todas estas mejoras incrementan en un 50 porciento de la eficiencia del algoritmo.

2.5 Métodos Aproximados

2.5.1 Wang (1983)[115]

En 1983 $Wang$ propuso dos métodos combinatorios que generan patrones de corte guillotina generales con restricciones sobre el número de veces que una pieza puede aparecer en un patrón de corte. Los métodos combinatorios realizan sucesivas uniones verticales y horizontales de los rectángulos $R_i = (l_i, w_i)$ demandado cada uno de los algoritmos propuesto utiliza un parámetro como una Cota sobre el máximo nivel que desperdicio aceptable que se cree durante el proceso de corte.

Una Unión vertical de los rectángulos $R_i = (p_1, q_1)\, yR_2 = (p_2, q_2\ )$ es un rectángulo S_v, de dimensiones $\{\max(p_1, p_2), (q_1 + q_2)\}$ qué contiene. R_1 y R_2. Una Unión horizontal $R_1 y$ R_2 de es un rectángulo S_v qué tiene dimensiones $\{p_1 + p_2, \max(q_1, q_2)\}$y contiene R_1, R_2; también se requiere que S_u y S_v no exceda las dimensiones del rectángulo original.

$Wang$ propuso lo siguiente algoritmo para resolver el problema de corte.

Algoritmo I

1. Seleccionar un valor para máximo valor desperdicio aceptable.

$\beta_1,\ 0 \leq \beta_1 \leq 1$

2. Definir conjunto de rectángulos que se deben cortar y hacer $K = 1\ L^{(0)} = F^{(0)}\{R_0, R_1.....R_n\}$,

hacer, $K\ = 1$

Etapa2.

1. Calcular $F^{(K)}$es el conjunto de todos los rectángulos T que satisfacen.

a. T está formado por uniones verticales y horizontales de dos rectángulos de $L^{(k-1)}$

b. La cantidad de desperdicio T en no debe exceder $\beta_1 HW$,

c. Todos los rectángulos R_i que aparecen en T cumplen las restricciones de cota $b_1, b_2 \ldots . b_{n.}$

2. Sea $L^{(k)} = L^{(k-1)}\ UF^{(k)}$ eliminar patrón es equivalente en $L^{(k)}$

Etapa3.

1. Si $F^{(k)}$ no es vacío, hacer $K = k + 1$ ir a la etapa2, en otro caso ir a la etapa 4

Etapa 4.

1. Hacer $M = K - 1$

2. seleccionar el rectángulo en $L^{(M)}$qué tiene el menor desperdicio dentro del tablero (H, W)

Algoritmo II

Se obtiene el algoritmo II realizando los siguientes cambios en el algoritmo I

a. Reemplazar etapa1.1 β_1 por β_2

b. Reemplazar etapa 2.1.2 por "La cantidad de desperdicio e T no excede $\beta_2 \alpha\,(T)$ es la superficie de T"

2.5.2 Morabito y Aranales (1996)[89]

En 1996 *R. Morabito y M. Arenales* desarrollaron una extensión del algoritmo $AND/OR - GRAPH$ propuesto por para resolver el problema

de corte de guillotina restringido. Los autores definen de forma trivial una Cota inferior sencilla en cada nodo N del grafo AND/OR-GRAPH usando patrones de corte con solamente una pieza y obtienen una Cota Superior considerando la relajación de un problema lineal en cada en cada nodo.

En estrategia de búsqueda utiliza un método híbrido descrito $Morabito$ (1994) [89] que combina la estrategia $Back_{Tracking}\,(BT)$ y $Hillclimbung$ (HC). El algoritmo $BT-HC$ no requiere mucha memoria computacional ya que solamente almacena el mejor camino conocido.
La extensión propuesta del método $AND/OR-GRAPH$ halla soluciones de calidad en pequeños tiempos de computación para el problema de corte guillotina general restringido.

2.5.3 Fayard et al. (1998) [46]

En 1998 $D.Fayard\ M.\ Hifi$ desarrollaron un heurístico qué trata con diversas versiones del problema TDC y está basado en el problema $Knapsach$ unidimensional que resuelve con técnicas de programación dinámica.
El método es una generalización de $Best\ Strips\ Cutting$ (BSC) propuesto por $Fayard\ y\ Zissmopoulos$ (1995) [48] ,para resolver problemas $(UUTDC)$ los autores lo llaman $General\ Best\ Strip\ Cutting\,(CBSC)$.
El procedimiento propuesto usa algunas propiedades interesantes del programa dinámico qué permiten extender la idea desarrollada por $Gilmore\ y\ Gomory$ a todos los rectángulos que se producen después de un corte, sin adición un

substancial esfuerzo computacional.

Los procedimientos $SEP, FP,\ y\ DP$ del algoritmoBSC se utilizan en este nuevo método con algunas modificaciones. Algunas suposiciones necesarias para tratar con las diferentes versiones del problema TDC Son las siguientes:

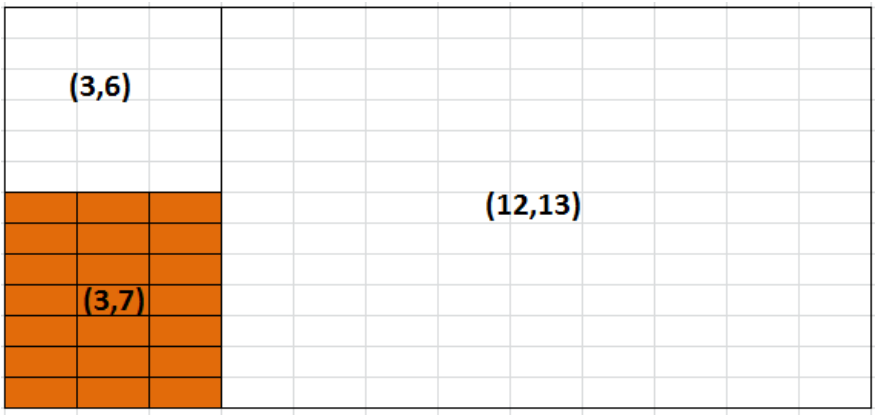
a. versión no restringida: por razones de simetría los conjuntos $P_L\ y\ P_W$ se restringen solamente a elementos más pequeños que $\frac{L}{2} y \frac{W}{2}$ respectivamente.
b. Versión restringida: se consideran todos los puntos de los conjuntos $P_L\ y\ P_H$ ya que no es válida la simetría.

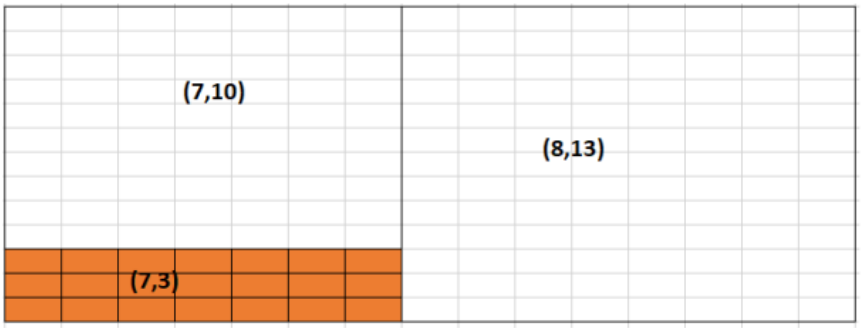
CAPITULO 3: ALGORITMOS HEURISTICOS

En esta sesión proponemos los algoritmos heurísticos constructivos de piezas con orientación fija y no fija, donde la pieza $(a, b) \neq (b, a)$, además de un constructivo basado en cortes en tiras o franjas para tratar con las cuatro versiones del problema TDC. Estos procedimientos constructivos serán usados como etapas en algoritmos más complejos, como el Tabu Search.

3.1 ALGORITMOS CONSTRUCTIVOS DE PIEZAS CON ORIENTACIÓN FIJA Y ROTADA

Se desarrollará un proceso de corte en el cual para cada sub rectángulo acortar consideremos la consecuencia de cortar cada pieza posible en el extremo inferior izquierdo del rectángulo. Considerando poner la piza en la esquina inferior izquierda de forma fija y rotada





3.1.2 Cota Superior $\mathbf{BK}_j$

Si tenemos que cortar una pieza l_i, w_i en el extremo inferior izquierdo del rectángulo. $R_k = (L_k, W_k)$ se tendrá una de estas dos situaciones: si el primer corte de guillotina es vertical, además de la pieza i obtendremos los dos rectángulos.

${R_1}^i = (l_i, W_k - w_i)$ y ${R_2}^i = (l_k - l_i\ , W_k)$, si el primer corte guillotina es horizontal, entonces además de la pieza i obtenemos $R_3^i = (l_k - l_i\ , w_i)$ y ${R_4}^i = (l_k, W_k - w_i)$.

Si el primer corte guillotina es horizontal, entonces además de la pieza i obtendremos:

$$BK_1(R) = Max \sum_{i \in S^*} v_i x_i$$

Donde $S^* = \{i / l_i \leq L,\ w_i \leq W\}$

$$s.t \sum_{i \in S^*} v_i x_i \leq LW$$

$0 \leq x_i \leq \ min \ \left\{ d_i - n_i, \left[\frac{L}{l_i}\right] \left[\frac{W}{w_i}\right] \right\} \ \ i \ = \ 1, \ldots..m$

Donde n_i es el número de piezas del tipo i ya cortadas, incluyendo la pieza que se está considerando. El problema $\mathrm{BK}_1\,(R)$ se resuelve de forma aproximada, usando el siguiente algoritmo Greedy propuesto por Martello y Toth (1990) [85] .

Etapa0. Inicialización

Sea $Z = 0$ el valor total de las piezas cortadas

$c = LW$, el valor actual del término independiente de las restricciones

El conjunto de piezas S^* se ordena de forma no creciente $\frac{v_i}{s_i}$

Sea $j = 1$ el índice de piezas que se considera actualmente

Sea $Z^* = 0$ el valor que se obtiene si solamente se corta un tipo de pieza y j^* su índice correspondiente.

Etapa1.

$u_j = min \left\{ d_j - n_j, \ \left[\frac{L}{l_j}\right] \left[\frac{W}{w_j}\right] \right\}$

$x_j = min \ \left\{ v_j, \frac{c}{s_j} \right\}$

$z = z + v_j x_j$

$c = c - s_j x_j$

Si $v_j u_j > z^*$, entonces $z^* = v_j u_j$ y $j^* = j$

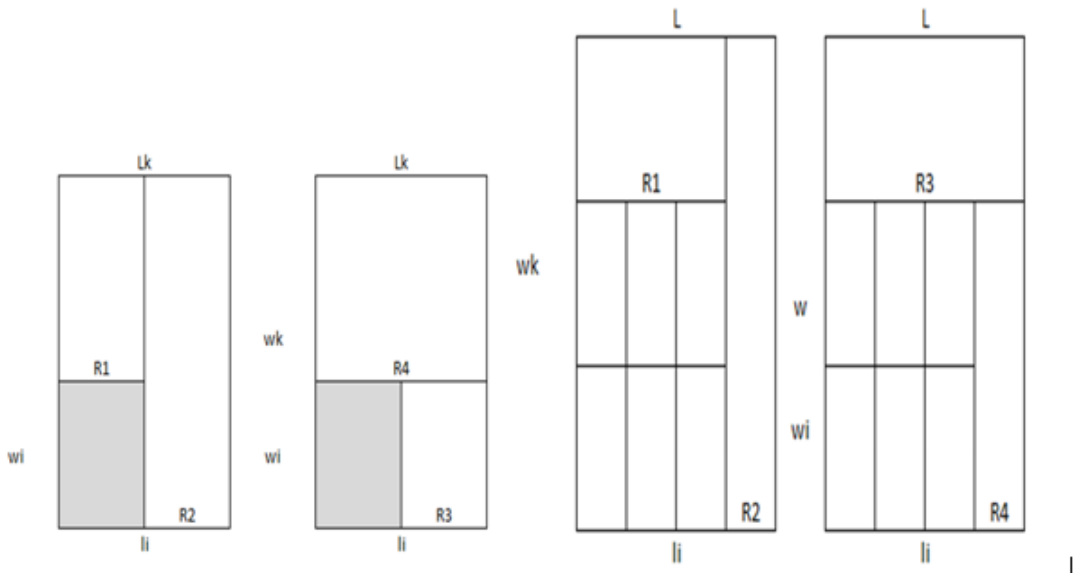
Si $j < m$, entonces $j \ = j + 1$ y volver a la etapa1.

En caso contrario ir a Etapa 3

Etapa3.

Si $z > z$, entonces para $j = 1 \ , \ldots..m, x_j = 0 \ ; x_j = u_j$

En base a esta estimación se deciden la pieza cortar hasta que no hallan más piezas que se puedan obtener de los rectángulos sobrantes los cuales se consideran desperdicio.



La idea de resolver dos problema de Knapsack para las regiones R_1 y R_2 en las figuras a o en las regiones R_3 yR_4 en la figura fue introducido por Hifi y Ouafi (1997)[70] .Para obtener cotas inferior en cada nodo de un algoritmo Branch and Bound ellos resuelven exactamente el problema de knapsack para las regiones grandes usando programación dinámica y suman el valor de la mejor solución homogénea de la región más pequeña, esto es la mejor solución que involucre un tipo de pieza.

Supongamos que usamos el algoritmo de greedy de Martello y Toth (1990) [85] para cortar el rectángulo $R = (L, W)$ y que en la etapa1 la primera pieza se corta x_i veces.

Cuando consideramos la segunda pieza, el lado derecho de la restricción será $LW - x_1 l_1 w_1$ y se intentará de cortar tantas piezas como sea posible sin exceder este límite.

Sin embargo, después de cortar la pieza 1 la situación puede ser como en la $x_i = 6$ por ejemplo, tomando en cuenta la forma en que la pieza se ha cortado determinamos n_i *y* n_w cotas inferiores sobre el número de veces que la pieza se ha Cortado, estos valores son tales como $n_i n_w = x_1$ para problemas no restringidos por lo tanto la situación en el ejemplo es $n_l = 3$, $n_w = 2$.

El espacio que sobra será R_1 y R_2, puede ser R_3 y R_4 donde:

$R_1 = (n_l l_i, W - n_w w_1)$, $R_2 = (L - n_l l_1, W)$

$R_3 = (L, W - n_w w_1)$, $R_4 = (L - n_l l_1, n_w w_1)$

Por lo tanto, para la pieza $i = 2, 3 \ldots . m$ su cota superior será:

$$x_i \leq min \left\{ d_i - n_i,\ max \left\{ \left[\frac{L_1}{l_i}\right] \left[\frac{W_1}{w_i}\right] + \left[\frac{L_2}{l_i}\right] \left[\frac{W_2}{w_2}\right], \left[\frac{L_3}{l_i}\right] \left[\frac{W_3}{w_i}\right] + \left[\frac{L_4}{l_i}\right] \left[\frac{W_4}{w_i}\right] \right\} \right\}$$

Donde $l_i w_i$ son las dimensiones del rectángulo $R_j,\ j = 1 \ldots .4$

En esta segunda cuota superior otra vez usando el algoritmo de Martello y Toth (1990)[85] pero desde $j = 2$ en adelante la Cota superior de cada pieza será la expresión de arriba está modificación tiene efecto significativo sobre las cotas la cosa de mesa grande y efectos pequeño sobre las gotas de piezas pequeñas.

3.1.2 El procedimiento Constructivo es el siguiente:

Etapa 0 Inicialización: Sea $L = \{R\}$ la lista de rectángulos aun por cortarse (inicialmente la lista contiene el rectángulo original R)

Hacer $P = \emptyset$ el conjunto de piezas ya cortadas

Hacer $v_T = 0$ el valor total de las piezas cortadas

El conjunto S de piezas a cortar es ordenado en orden no creciente de

$r_1 = \frac{vi}{si}$ valor partido por la superficie. Los empates se rompen por el orden

decreciente de v_i para cada pieza i, n_i es el número de piezas cortadas

inicialmente, $n_i = 0$

Etapa1. Mientra $L \neq \emptyset$

Tomar el rectángulo más pequeño de $L, R_K = (L_K, W_K)$ y hacer

$L = L - \{R_K\}$

Etapa 2. Para cada pieza. $i,\ i = 1, \ldots m$

Si $l_i \leq L_K,\ w_i \leq w_k,\ n_i \leq d_i$

Considerar las consecuencias de cortar la pieza i en el extremo inferior

izquierdo de R_K.

Calcular las estimaciones

$e_1 = Bk\left({R_1}^k\right)$, donde ${R^K}_1 = (l_i, W_k - w_i)$ y

$e_2 = Bk\left({R_2}^k\right)$, donde ${R^K}_2 = (l_k - l_i, W_k)$

Obteniendo por un corte guillotina horizontal también calcular

$h_1 = Bk\left({R_3}^k\right)$, donde $R_3^K = (l_k - l_i, w_i)$ y

$h_4 = Bk\left({R_4}^k\right)$, donde ${R^K}_4 = (l_k, W_k - w_i)$

Obteniendo por un corte de guillotina horizontal

El posible beneficio total de cortar una pieza i es:

$B_i = v_i + max\{e_1 + e_2, h_1 + h_2\}$

En caso contrario (la pieza no cabe en el rectángulo)

$B_i = 0$

Etapa 3. para la pieza j tal que $B_j = maxi\{B_i\}$

Si $B_j > 0$ Cortar la pieza j en el extremo inferior izquierdo de $R_{k.}$

Hacer $P = P \cup \{j\}, n_j = n_j + 1$ y $v_T = v_T + v_j$

Si $e_1 + e_2 \geq h_1 + h_2$

$L = L \cup \left\{R_1{}^j\right\} \cup \left\{R_2{}^j\right\}$

En caso contrario

$L = L \cup \left\{R_3{}^j\right\} \cup \left\{R_4{}^j\right\}$

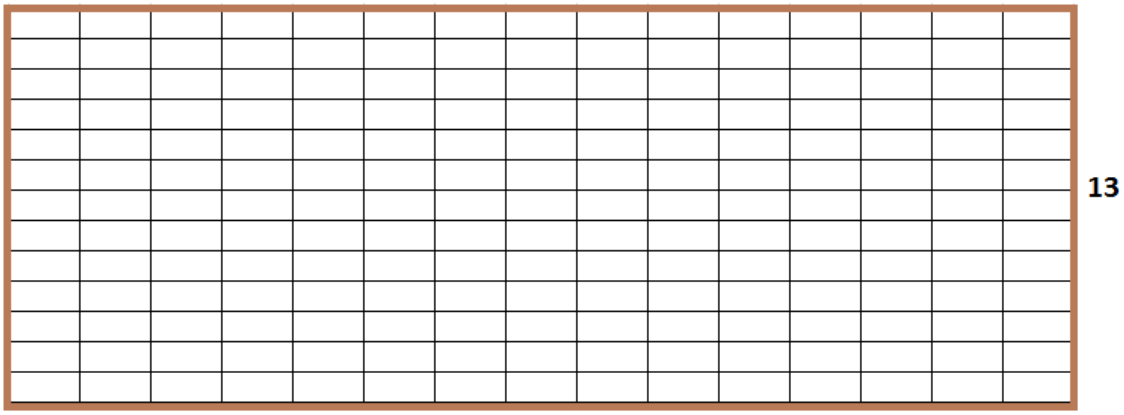
Si $B_j = 0$ ninguna pieza cabe en el rectángulo.

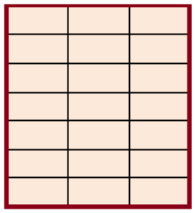
El rectángulo R_K es el desperdicio ir a la etapa1.

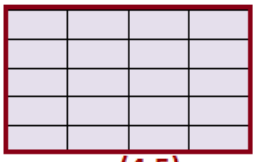
Notar que si en la Etapa2 en vez de BK_1 se usa la cota superior BK_2, se tiene otro algoritmo heurístico.

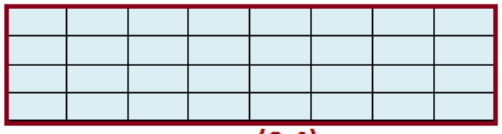
Para problema de corte, el tratamiento metodológico que utilizaremos es tomar un tablero bidimensional en pieza más pequeña, tal manera que se optimice su beneficio y minimizar el desperdicio.

Ejemplo: Dado un tablero cuyas dimensiones son 15 u de largo y 13 u de ancho. Se desea cortar en piezas más pequeñas, estas piezas deben de ser de dimensiones 3u x 7u, 4u x 5u y 8u x 4u. Para determinar los tipos y cantidad de cortes que se deben hacer para optimizar los beneficios. A continuación, se observa el tablero por cortar y los tipos de piezas demandadas.









i	(l_i, w_i)	v_i	s_i	$r_i = \frac{v_i}{s_i}$	d_i	$\left\lfloor \frac{L}{l_i} \right\rfloor \left\lfloor \frac{W}{w_i} \right\rfloor$
1	(3,7)	78	21	3.72	4	$\left\lfloor \frac{15}{3} \right\rfloor \left\lfloor \frac{10}{7} \right\rfloor = 5$
2	(8,4)	65	32	2.03	6	$\left\lfloor \frac{15}{8} \right\rfloor \left\lfloor \frac{10}{4} \right\rfloor = 2$
3	(4,5)	28	20	1.40	10	$\left\lfloor \frac{15}{4} \right\rfloor \left\lfloor \frac{10}{75} \right\rfloor = 6$

COTA SUPERIOR BK1

ETAPA 0

$R = \{(15, 13)\}$ *Rectángulo original*

$P = \{\} = \phi$ *Conjunto de piezas cortadas*

$V_T = 0$ *Valor total de piezas cortadas*

$S = li, wi : i = 3, 7, 8, 4, 4, 5$ *Piezas que se cortar*

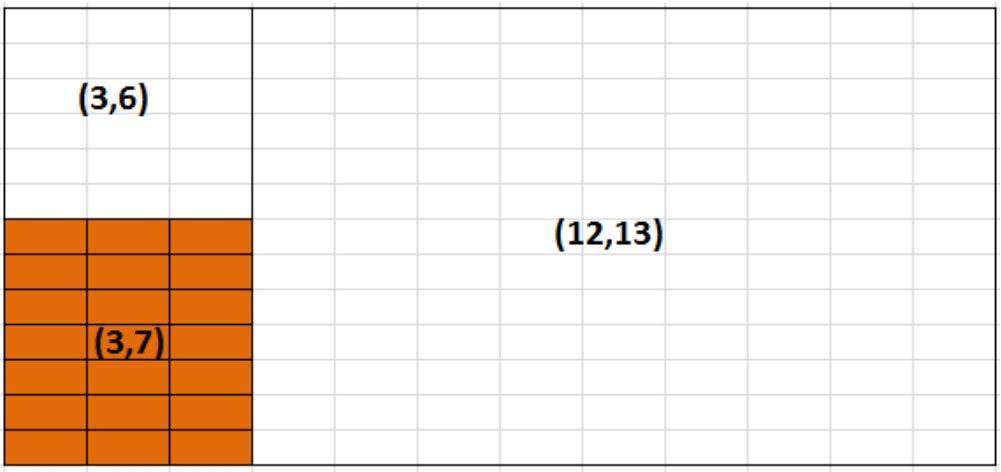
$n_1 = n_2 = n_3 = 0$ *Número de piezas de tipo*

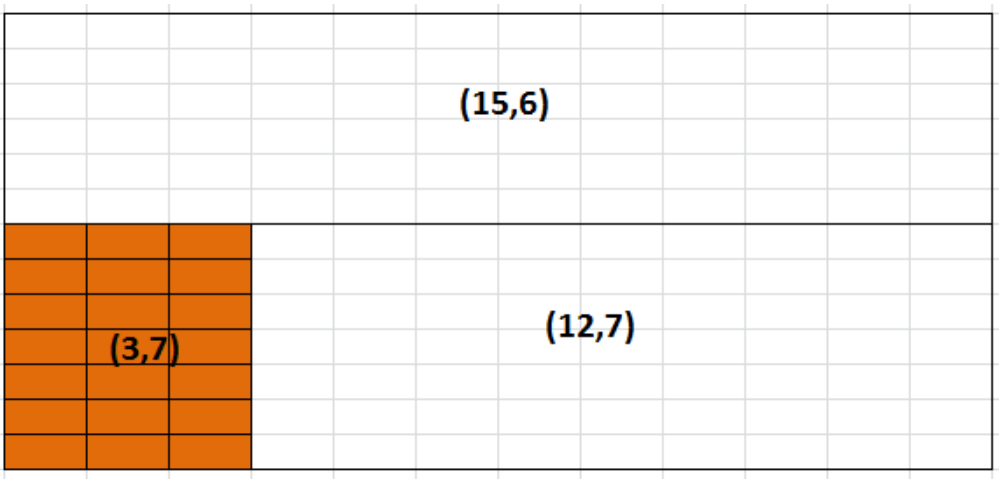
ETAPA 1

$R_k = \{(15, 13)\}$

$\{(3, 7), (8, 4), (4, 5)\}$ *Piezas que se pueden cortar en* R_k

Pieza $(\mathbf{3}, \mathbf{7})$





Calcular los estimadores de los 4 tableros resultantes:

$e_1 = B_{k1}(3,6) =?$

$e_2 = B_{k1}(12,13) =?$

$h_1 = B_{k1}(12,7) =?$

$h_2 = B_{k1}(15,6) =?$

Para la primera estimacion vertical tenemos:

$e_1 = B_{k1}(3,6) = 0, S^* = \phi,$ *no se puede cortar ninguna pieza en* $(3,6)$

Para la segunda estimacion vertical tenemos:

$e_1 = B_{k1}(12,13)$

$S^* = \{(3,7),(8,4),(4,5)\}$

Resolvemos problema Knapsack

$Max\ 78X_1 + 65X_2 + 28X_3$

S.a $21X_1 + 32X_2 + 20X_3 \leq 156$

$0 \leq X_1 \leq min\left\{6-0,\ \left\lfloor\frac{12}{3}\right\rfloor \left\lfloor\frac{13}{7}\right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6,4\} \Longrightarrow 0 \leq X_1 \leq 4$

$0 \leq X_2 \leq min\left\{8-0,\ \left\lfloor\frac{12}{8}\right\rfloor \left\lfloor\frac{13}{4}\right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\{8,3\} \Longrightarrow 0 \leq X_2 \leq 3$

$0 \leq X_3 \leq min\left\{10-0,\ \left\lfloor\frac{12}{4}\right\rfloor \left\lfloor\frac{13}{5}\right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10,6\} \Longrightarrow 0 \leq X_3 \leq 6$

Hacemos:

$U_1 = min\{6,4\} = 4 \Longrightarrow (4)(78) = 312 \Longrightarrow Z^* = 312,\ j^* = 1$

$U_2 = min\{8,3\} = 3 \Longrightarrow (3)(195) = 130312 \Longrightarrow Z^* = 312,\ j^* = 1$

$U_3 = min\{10,6\} = 6 \Longrightarrow (6)(28) = 168312 \Longrightarrow Z^* = 312,\ j^* = 1$

Hacemos cortes sucesivos

$X_1 \leq min\left\{4, \left\lfloor\frac{156}{21}\right\rfloor\right\} = 4$

$Z = (4)(78) = 312$

$C = 156 - (4)(21) = 72$

$X_2 \leq min\left\{3, \left\lfloor \frac{72}{32} \right\rfloor\right\} = 2$

$Z = 312 + (2)\,(65) = 442$

$C = 72 - (2)(32) = 8$

$X_3 \leq min\left\{6, \left\lfloor \frac{8}{20} \right\rfloor\right\} = 0$

$Z = 442 + 0 = 442$

$C = 8$

Luego:

$Z^* = 312442 \Longrightarrow Z = 442,\ X_1 = 4,\ X_2 = 2,\ X_3 = 0$

$e_2 = B_{k1}(12, 13) = 442$

. Para estimar de manera horizontal tenemos:

$h_1 = B_{k1}(12, 7)$

$S^* = \{(3, 7)\,, (8, 4)\,, (4, 5)\}$

Resolvemos problema Knapsack

$Max\ 78X_1 + 65X_2 + 28X_3$

S.a $21X_1 + 32X_2 + 20X_3 \leq 84$

$0 \leq X_1 \leq min\left\{6 - 0,\ \left\lfloor \frac{12}{3} \right\rfloor \left\lfloor \frac{7}{7} \right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6, 4\} \Longrightarrow 0 \leq X_1 \leq 4$

$0 \leq X_2 \leq min\left\{8 - 0,\ \left\lfloor \frac{12}{8} \right\rfloor \left\lfloor \frac{7}{4} \right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\{8, 1\} \Longrightarrow 0 \leq X_2 \leq 1$

$0 \leq X_3 \leq min\left\{10 - 0,\ \left\lfloor \frac{12}{4} \right\rfloor \left\lfloor \frac{7}{5} \right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10, 3\} \Longrightarrow 0 \leq X_3 \leq 3$

Hacemos:

$U_1 = min\{6, 4\} = 4 \Longrightarrow (4)\,(78) = 312 \Longrightarrow Z^* = 312,\ j^* = 1$

$U_2 = min\{8, 1\} = 1 \Longrightarrow (1)\,(65) = 65312 \Longrightarrow Z^* = 312,\ j^* = 1$

$U_3 = min\{10, 3\} = 3 \Longrightarrow (3)\,(28) = 84312 \Longrightarrow Z^* = 312,\ j^* = 1$

Hacemos cortes sucesivos:

$X_1 \leq min\left\{4, \left\lfloor \frac{84}{21} \right\rfloor\right\} = 4$

$Z = (4)\,(78) = 312\ C = 84 - (4)(21) = 0$

$X_2 \leq min\left\{1, \left\lfloor \frac{0}{32} \right\rfloor\right\} = 0$

$Z = 312 + (0)\,(65) = 312\ C = 0$

$X_3 \leq min\left\{3, \left\lfloor \frac{0}{20} \right\rfloor\right\} = 0$

$Z = 312 + (0)\,(28) = 312$

$C = 0$

Luego:

$Z^* = 312312 \Longrightarrow Z = 312,\ X_1 = 4,\ X_2 = 0,\ X_3 = 0$

$h_1 = B_{k1}(12, 7) = 312$

. Para estimar de manera horizontal tenemos:

$h_2 = B_{k1}(15, 6)$

$S^* = \{(8, 4)\,, (4, 5)\}$

Resolvemos problema Knapsack

$Max\ 65X_2 + 28X_3$

S.a $32X_2 + 20X_3 \leq 90$

$0 \leq X_2 \leq min\left\{6 - 0,\ \left\lfloor \frac{15}{8} \right\rfloor \left\lfloor \frac{6}{4} \right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\{6, 1\} \Longrightarrow 0 \leq X_2 \leq 1$

$0 \leq X_3 \leq min\left\{10 - 0,\ \left\lfloor \frac{15}{4} \right\rfloor \left\lfloor \frac{6}{5} \right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10, 3\} \Longrightarrow 0 \leq X_3 \leq 3$

Hacemos:

$U_2 = min\{6, 1\} = 1 \Longrightarrow (1)\,(65) \Longrightarrow Z^* = 65,\ j^* = 2$

$U_3 = min\{10, 3\} = 3 \Longrightarrow (3)\,(28) = 84 > 65 \Longrightarrow Z^* = 84,\ j^* = 3$

Hacemos cortes sucesivos:

$X_2 \leq min\left\{1, \left\lfloor \frac{90}{32} \right\rfloor\right\} = 1$

$Z = (1)(65) = 65$

$C = 90 - 1(32) = 58$

$X_3 \leq min\left\{3, \left\lfloor \frac{58}{20} \right\rfloor\right\} = 2$

$Z = 65 + (2)(28) = 121$

$C = 58 - 40 = 18$

Luego:

$Z^* = 84121 \Longrightarrow Z = 121,\ X_1 = 0,\ X_2 = 1,\ X_3 = 2$

$h_2 = B_{k1}(12, 10) = 121$

$e_1 = B_{k1}(3, 6) = 0$

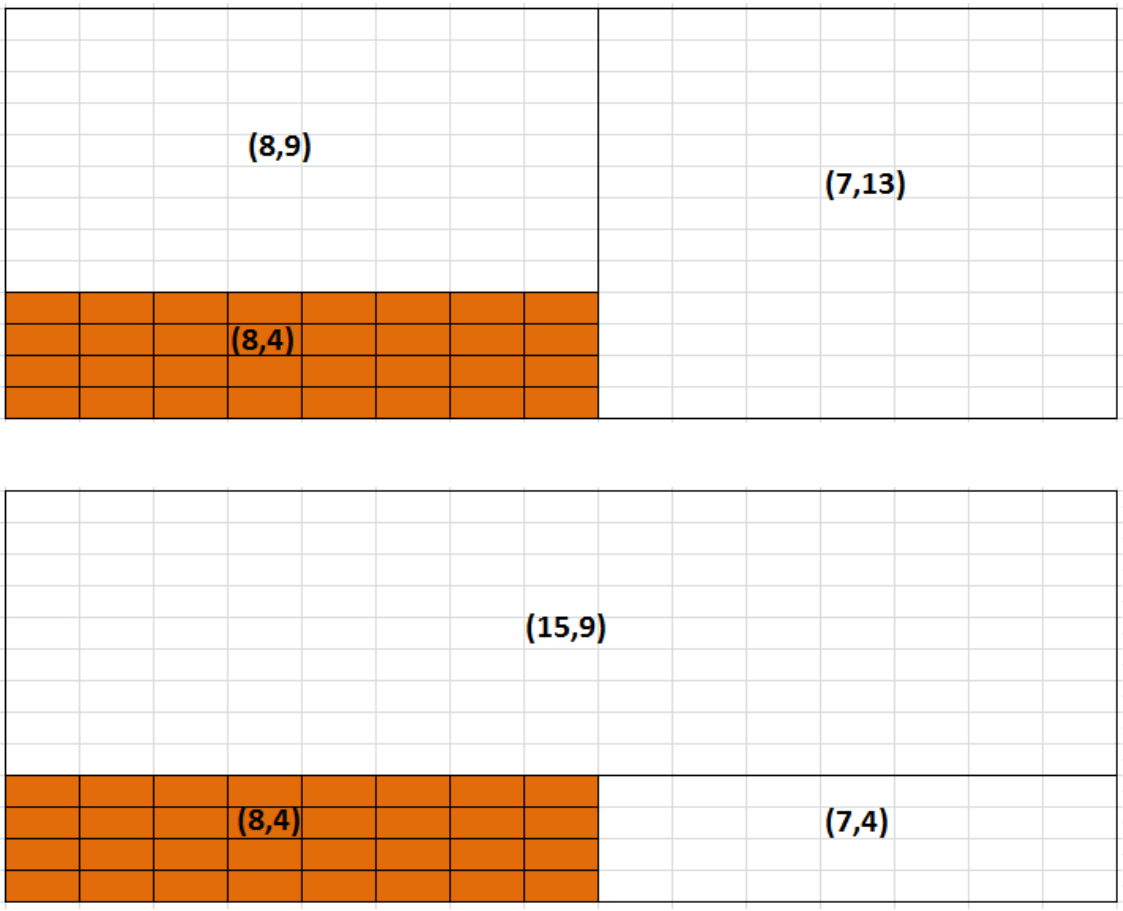
$e_2 = B_{k1}(12, 13) = 442$

$h_1 = B_{k1}(12, 7) = 312$

$h_2 = B_{k1}(15, 6) = 121$

PIEZA	CORTE	ESTIMADOR	SUMA	$B_i = V_i + max\{e_1 + e_1, h_1 + h_{12}\}$
(3,7)	VERTICAL	0	442	78+442=520
		442		
	HORIZONTAL	312	433	
		121		

Pieza $(\mathbf{8}, \mathbf{4})$



Calcular los estimadores de los 4 tableros resultantes.

$e_1 = B_{k1}(8, 9) =?$

$e_2 = B_{k1}(7, 13) =?$

$h_1 = B_{k1}(7, 4) =?$

$h_2 = B_{k1}(15, 9) =?$

Para la primera estimacion vertical tenemos:

$e_1 = B_{k1}(8, 9)$

$\mathrm{S}^* = \{(3, 7), (8, 4), (4, 5)\}$

Resolvemos problema Knapsack

$Max\ 78X_1 + 65X_2 + 28X_3$

S.a $21X_1 + 32X_2 + 20X_3 \leq 72$

$0 \leq X_1 \leq min\left\{6-0,\ \left\lfloor\frac{8}{3}\right\rfloor \left\lfloor\frac{9}{7}\right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\left\{6,2\right\} \Longrightarrow 0 \leq X_2 \leq 2$

$0 \leq X_2 \leq min\left\{8-0,\ \left\lfloor\frac{8}{8}\right\rfloor \left\lfloor\frac{9}{4}\right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\left\{8,2\right\} \Longrightarrow 0 \leq X_2 \leq 2$

$0 \leq X_3 \leq min\left\{10-0,\ \left\lfloor\frac{8}{4}\right\rfloor \left\lfloor\frac{9}{5}\right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\left\{10,2\right\} \Longrightarrow 0 \leq X_3 \leq 2$

Hacemos:

$U_1 = min\left\{6,2\right\} = 2 \Longrightarrow (2)\,(78) = 156 \Longrightarrow Z^* = 156,\ j^* = 1$

$U_2 = min\left\{8,2\right\} = 2 \Longrightarrow (1)\,(65) = 130156 \Longrightarrow Z^* = 156,\ j^* = 1$

$U_3 = min\left\{10,2\right\} = 2 \Longrightarrow (2)\,(28) = 56156 \Longrightarrow Z^* = 156,\ j^* = 1$

Hacemos cortes sucesivos

$X_1 \leq min\left\{2, \left\lfloor\frac{72}{21}\right\rfloor\right\} = 2$

$Z = (2)\,(78) = 156$

$C = 72 - (2)(21) = 30$

$X_2 \leq min\left\{2, \left\lfloor\frac{30}{32}\right\rfloor\right\} = 0$

$Z = 156 + (0)\,(65) = 156$

$C = 30 - (0)(32) = 30$

$X_3 \leq min\left\{2, \left\lfloor\frac{30}{20}\right\rfloor\right\} = 1$

$Z = 156 + 1(28) = 184$

$C = 30 - 20 = 10$

Luego:

$Z^* = 156184 \Longrightarrow Z = 84,\ X_1 = 2,\ X_2 = 0,\ X_3 = 1$

$e_1 = B_{k1}(8,6) = 6$

Primera estimacion vertical:

$e_2 = B_{k1}(7, 13)$

$S^* = \{(3, 7), (4, 5)\}$

Resolvemos problema Knapsack

$Max\ 78X_1 + 28X_3$

S.a $21X_1 + 20X_3 \leq 91$

$0 \leq X_1 \leq min\left\{6 - 0,\ \left\lfloor \frac{7}{3} \right\rfloor \left\lfloor \frac{13}{7} \right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6, 2\} \Longrightarrow 0 \leq X_1 \leq 2$

$0 \leq X_3 \leq min\left\{10 - 0,\ \left\lfloor \frac{7}{4} \right\rfloor \left\lfloor \frac{13}{5} \right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10, 2\} \Longrightarrow 0 \leq X_3 \leq$

2 Hacemos:

$U_1 = min\{4, 2\} = 2 \Longrightarrow (2)(78) = 156 \Longrightarrow Z^* = 156,\ j^* = 1$

$U_3 = min\{10, 2\} = 2 \Longrightarrow (2)(28) = 56156 \Longrightarrow Z^* = 156,\ j^* = 1$

Hacemos cortes sucesivos:

$X_1 \leq min\left\{2, \left\lfloor \frac{91}{21} \right\rfloor\right\} = 2$

$Z = (2)(78) = 156$

$C = 91 - (2)(21) = 49$

$X_3 \leq min\left\{2, \left\lfloor \frac{49}{20} \right\rfloor\right\} = 2$

$Z = 156 + (2)(28) = 212$

$C = 49 - 40 = 9$

Luego:

$Z^* = 156212 \Longrightarrow Z = 212,\ X_1 = 2,\ X_2 = 0,\ X_3 = 2$

$e_2 = B_{k1}(7, 13) = 212$

Para estimacion horizontal tenemos:

$h_1 = B_{k1}(7,4) = 0$,S$^* = \phi$, *no se puede cortar ninguna pieza en* $(3,6)$

$h_2 = B_{k1}(15,9)$

$S^* = \{(3,7),(8,4),(4,5)\}$

Resolvemos problema Knapsack:

$Max\ 78X_1 + 65X_2 + 28X_3$

$0 \leq X_1 \leq min\left\{6-0,\ \left\lfloor\frac{15}{3}\right\rfloor \left\lfloor\frac{9}{7}\right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6,5\} \Longrightarrow 0 \leq X_2 \leq 5$

$0 \leq X_2 \leq min\left\{8-0,\ \left\lfloor\frac{15}{8}\right\rfloor \left\lfloor\frac{9}{4}\right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\{6,2\} \Longrightarrow 0 \leq X_2 \leq 2$

$0 \leq X_3 \leq min\left\{10-0,\ \left\lfloor\frac{15}{4}\right\rfloor \left\lfloor\frac{9}{5}\right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10,3\} \Longrightarrow 0 \leq X_3 \leq$

3

Hacemos:

$U_1 = min\{6,5\} = 5 \Longrightarrow (5)(78) = 390 \Longrightarrow Z^* = 390,\ j^* = 1$

$U_2 = min\{8,2\} = 2 \Longrightarrow (2)(65) = 130390 \Longrightarrow Z^* = 390,\ j^* = 1$

$U_3 = min\{10,3\} = 3 \Longrightarrow (3)(28) = 84390 \Longrightarrow Z^* = 390,\ j^* = 1$

Hacemos cortes sucesivos:

$X_1 \leq min\left\{5, \left\lfloor\frac{135}{21}\right\rfloor\right\} = 5$

$Z = 5(78) = 390$

C=135-5(21) =30

$X_2 \leq min\left\{2, \left\lfloor\frac{30}{32}\right\rfloor\right\} = 0$

$Z = 390 + (0)(65) = 390$

$C = 30 - 0 = 30$

$X_3 \leq min\left\{3, \left\lfloor\frac{30}{20}\right\rfloor\right\} = 1$

$Z = 390 + (1)(28) = 418$

$C = 30 - 20 = 10$

Luego:

$Z^* = 390418 \Longrightarrow Z = 418,\ X_1 = 5,\ X_2 = 0,\ X_3 = 1$

$h_2 = B_{k1}\,(12, 10) = 418$

. Resultado de la estimacion :

$e_1 = B_{k1}(3, 3) = 65$

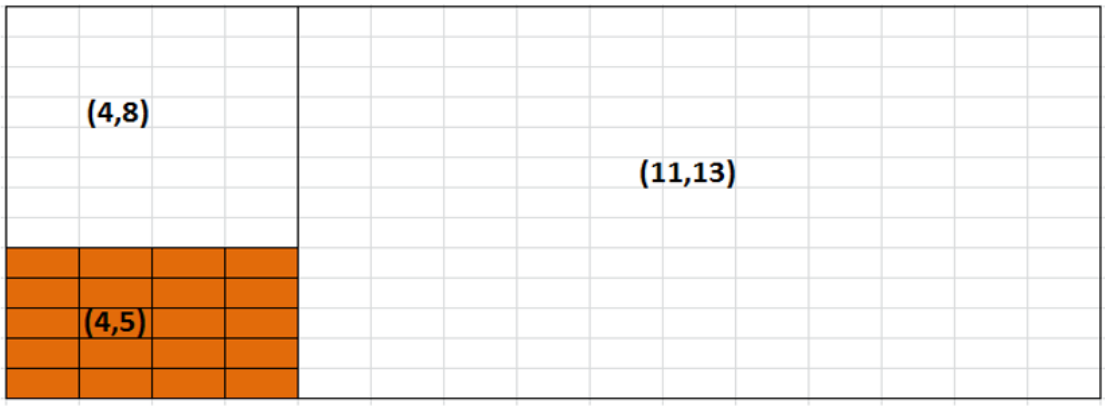
$e_2 = B_{k1}(12, 10) = 212$

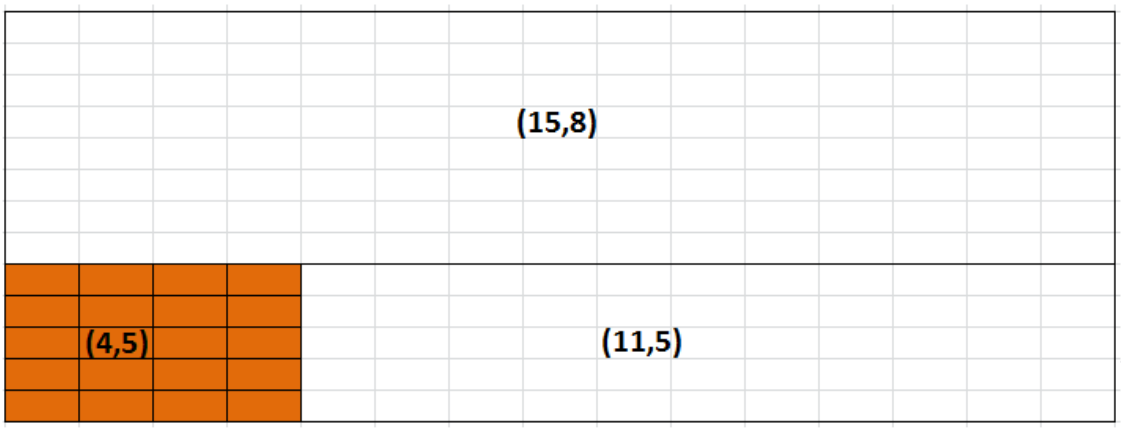
$h_1 = B_{k1}(12, 7) = 0$

$h_2 = B_{k1}(3, 3) = 418$

PIEZA	CORTE	ESTIMADOR	SUMA	$B_i = V_i + max\{e_1 + e_1, h_1 + h_{12}\}$
(8,4)	VERTICAL	65	277	65+418=483
		212		
	HORIZONTAL	0	418	
		418		

Pieza (4,5)





$e_1 = B_{k1}(4,8) = ?$

$e_2 = B_{k1}(11,13) = ?$

$h_1 = B_{k1}(15,8) = ?$

$h_2 = B_{k1}(11,5) = ?$

Primera estimacion vertical:

$e_1 = B_{k1}(4,8)$

$S^* = \{(3,7),(4,5)\}$

Resolvemos problema Knapsack:

$Max\ 78X_1 + 28X_3$

S.a $21X_1 + 20X_3 \leq 32$

$0 \leq X_1 \leq min\left\{6-0,\ \left\lfloor\frac{4}{3}\right\rfloor \left\lfloor\frac{8}{7}\right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6,1\} \Longrightarrow 0 \leq X_1 \leq 1$

$0 \leq X_3 \leq min\left\{10-0,\ \left\lfloor\frac{4}{4}\right\rfloor \left\lfloor\frac{8}{5}\right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10,1\} \Longrightarrow 0 \leq X_3 \leq 1$

Hacemos:

$U_1 = min\{6,1\} = 1 \Longrightarrow (1)(78) = 78 \Longrightarrow Z^* = 78,\ j^* = 1$

$U_3 = min\{10,1\} = 1 \Longrightarrow (1)(28) = 2878 \Longrightarrow Z^* = 78,\ j^* = 1$

Hacemos cortes sucesivos:

$X_1 \leq min\left\{1, \left\lfloor \frac{32}{21} \right\rfloor\right\} = 1$

$Z = (1)(78) = 78$

$C = 32 - (1)(21) = 11$

$X_3 \leq min\left\{1, \left\lfloor \frac{11}{20} \right\rfloor\right\} = 0$

$Z = 78 + (0)(28) = 78$

$C = 11 - 0 = 11$

Luego:

$Z^* = 7878 \Longrightarrow Z = 78,\ X_1 = 1,\ X_2 = 0,\ X_3 = 0$

$e_1 = B_{k1}(4, 8) = 78$

Estimar corte sucesivos vertical:

$e_2 = B_{k1}(11, 13)$

$S^* = \{(3, 7), (8, 4), (4, 5)\}$

Resolvemos problema Knapsack:

$Max\ 78X_1 + 65X_2 + 28X_3$

S.a $21X_1 + 32X_2 + 20X_3 \leq 110$

$0 \leq X_1 \leq min\left\{6 - 0,\ \left\lfloor \frac{11}{3} \right\rfloor \left\lfloor \frac{13}{7} \right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6, 3\} \Longrightarrow 0 \leq X_1 \leq 3$

$0 \leq X_2 \leq min\left\{8 - 0,\ \left\lfloor \frac{11}{8} \right\rfloor \left\lfloor \frac{13}{4} \right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\{8, 3\} \Longrightarrow 0 \leq X_2 \leq 3$

$0 \leq X_3 \leq min\left\{10 - 0,\ \left\lfloor \frac{11}{4} \right\rfloor \left\lfloor \frac{13}{5} \right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10, 4\} \Longrightarrow 0 \leq X_3 \leq 4$

Hacemos:

$U_1 = min\{6, 3\} = 3 \Longrightarrow (3)(78) = 234 \Longrightarrow Z^* = 234,\ j^* = 1$

$U_2 = min\{8, 3\} = 3 \Longrightarrow (3)(65) = 195234 \Longrightarrow Z^* = 234,\ j^* = 1$

$U_3 = min\{10, 4\} = 4 \Longrightarrow (4)(28) = 112234 \Longrightarrow Z^* = 234,\ j^* = 1$

Hacemos cortes sucesivos:

$X_1 \leq min\left\{3, \left\lfloor \frac{143}{21} \right\rfloor\right\} = 3$

$Z = (3)\,(78) = 234$

$C = 143 - (3)(21) = 80$

$X_2 \leq min\left\{3, \left\lfloor \frac{80}{32} \right\rfloor\right\} = 2$

$Z = 234 + (2)\,(65) = 364$

$C = 80 - (2)(32) = 16$

$X_3 \leq min\left\{4, \left\lfloor \frac{16}{20} \right\rfloor\right\} = 0$

$Z = 364 + (0)\,(28) = 364$

$C = 364 + 0 = 364$

Luego:

$Z^* = 234429 \Longrightarrow Z = 429,\ X_1 = 3,\ X_2 = 2,\ X_3 = 0$

$e_2 = B_{k1}(11, 10) = 299$

Estimar corte sucesivo vertical:

$h_1 = B_{k1}(15, 8)$

$S^* = \{(3, 7)\,, (8, 4)\,, (4, 5)\}$

Resolvemos problema Knapsack:

$Max\ 78X_1 + 65X_2 + 28X_3$

S.a $21X_1 + 32X_2 + 20X_3 \leq 120$

$0 \leq X_1 \leq min\left\{6 - 0,\ \left\lfloor \frac{15}{3} \right\rfloor \left\lfloor \frac{8}{7} \right\rfloor\right\} \Longrightarrow 0 \leq X_1 \leq min\{6, 5\} \Longrightarrow 0 \leq X_1 \leq 5$

$0 \leq X_2 \leq min\left\{8 - 0,\ \left\lfloor \frac{15}{8} \right\rfloor \left\lfloor \frac{8}{4} \right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\{8, 2\} \Longrightarrow 0 \leq X_2 \leq 2$

$0 \leq X_3 \leq min\left\{10 - 0,\ \left\lfloor \frac{15}{4} \right\rfloor \left\lfloor \frac{8}{5} \right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\{10, 3\} \Longrightarrow 0 \leq X_3 \leq 3$

Hacemos:

$U_1 = min\{6,5\} = 5 \Longrightarrow (5)(78) = 390 \Longrightarrow Z^* = 390,\ j^* = 1$

$U_2 = min\{8,2\} = 2 \Longrightarrow (2)(65) = 130390 \Longrightarrow Z^* = 390,\ j^* = 1$

$U_3 = min\{10,3\} = 3 \Longrightarrow (3)(28) = 84234 \Longrightarrow Z^* = 234,\ j^* = 1$

Hacemos cortes sucesivos:

$X_1 \leq min\left\{5, \left\lfloor \frac{120}{21} \right\rfloor\right\} = 5$

$Z = (5)(78) = 390$

$C = 120 - (5)(21) = 15$

$X_2 \leq min\left\{2, \left\lfloor \frac{15}{32} \right\rfloor\right\} = 0$

$Z = 390 + (0)(65) = 390$

$C = 15 - 0 = 15$

$X_3 \leq min\left\{3, \left\lfloor \frac{15}{20} \right\rfloor\right\} = 0$

$Z = 390 + (0)(28) = 390$

$C = 390 + 0 = 390$

Luego:

$Z^* = 390390 \Longrightarrow Z = 390,\ X_1 = 5,\ X_2 = 0,\ X_3 = 0$

$h_1 = B_{k1}(15,8) = 390$

Estimar corte sucesivos horizontal:

$h_2 = B_{k1}(11,5)$

$S^* = \{(8,4),(4,5)\}$

Resolvemos problema Knapsack:

$Max\ 65X_2 + 28X_3$

S.a $32X_2 + 20X_3 \leq 55$

$0 \leq X_2 \leq min\left\{8-0,\ \left\lfloor \frac{11}{8} \right\rfloor \left\lfloor \frac{5}{4} \right\rfloor\right\} \Longrightarrow 0 \leq X_2 \leq min\left\{6,1\right\} \Longrightarrow 0 \leq X_2 \leq 1$

$0 \leq X_3 \leq min\left\{10-0,\ \left\lfloor \frac{11}{4} \right\rfloor \left\lfloor \frac{5}{5} \right\rfloor\right\} \Longrightarrow 0 \leq X_3 \leq min\left\{10,2\right\} \Longrightarrow 0 \leq X_3 \leq 2$

Hacemos:

$U_2 = min\left\{8,1\right\} = 1 \Longrightarrow (1)(65) \Longrightarrow Z^* = 65,\ j^* = 2$

$U_3 = min\left\{10,2\right\} = 2 \Longrightarrow (3)(28) = 56 > 65 \Longrightarrow Z^* = 65,\ j^* = 3$

Hacemos cortes sucesivos:

$X_2 \leq min\left\{1, \left\lfloor \frac{55}{32} \right\rfloor\right\} = 1$

$Z = (1)(65) = 65$

$C = 55 - 1(32) = 23$

$X_3 \leq min\left\{2, \left\lfloor \frac{23}{20} \right\rfloor\right\} = 1$

$Z = 65 + (1)(28) = 93$

$C = 23 - 20 = 3$

Luego:

$Z^* = 6593 \Longrightarrow Z = 121,\ X_1 = 0,\ X_2 = 1,\ X_3 = 1$

$h_2 = B_{k1}(11,5) = 93$

Resultados de las estimaciones de los corte sucesivos:

$e_1 = B_{k1}(4,8) = 78$

$e_2 = B_{k1}(11,13) = 299$

$h_1 = B_{k1}(15,8) = 390$

$h_2 = B_{k1}(11,5) = 93$

PIEZA	CORTE	ESTIMADOR	SUMA	$B_i = V_i + max\{e_1 + e_1, h_1 + h_{12}\}$
(4,5)	VERTICAL	78	377	28+483=511
		299		
	HORIZONTAL	390	483	
		93		

ETAPA 3

$B = Max\{B_i,\ i \in I\}$

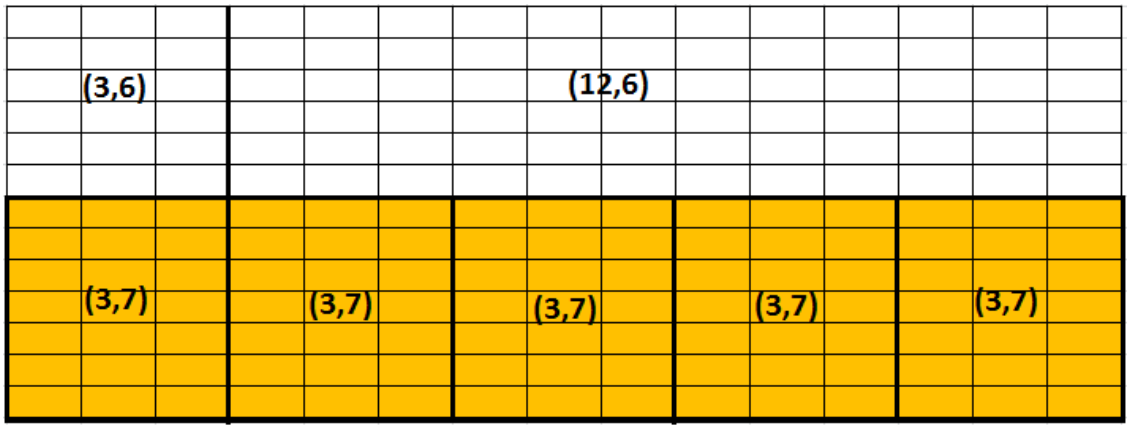
$B = Max\{520, 483, 511\} = 520$

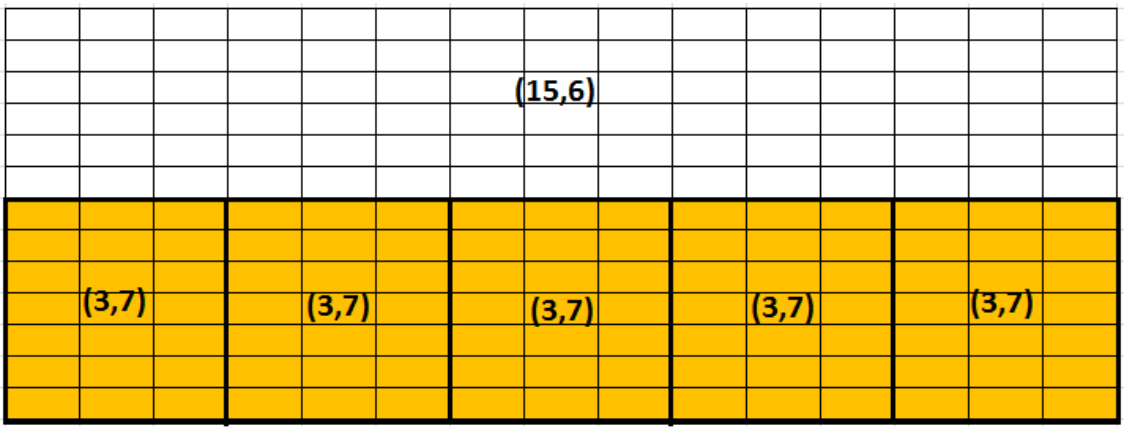
$B > 0 \Longrightarrow P = P \cup \{1\},\quad P = \{(3,7)\},\ n_1 = 0 + 1 = 1,\ V_T = 78$

$Si(e_1 + e_2)(h_1 + h_1)entoncesR = RU[(3,3)]U[(12,10]$

$en\ caso\ contrario\ R = R \cup \{(15,3)\} \cup \{12,7\}$

Por tanto, la pieza cortada es la (3,4) y los nuevos rectángulos a cortar son (3,3) y (12,10)

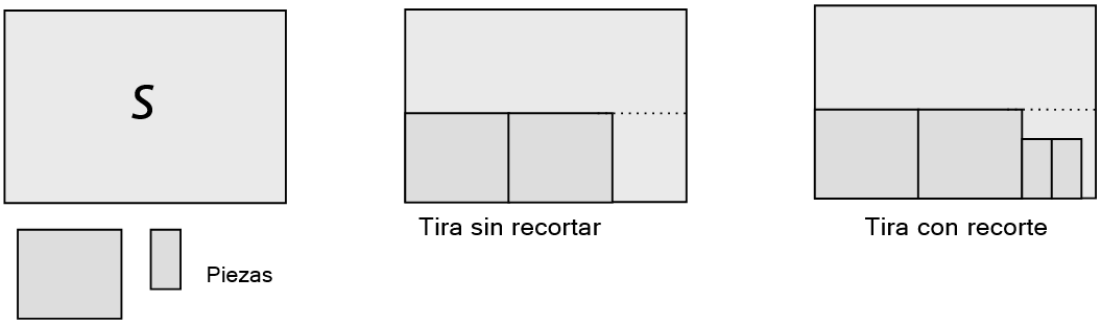




3.2. PROCEDIMIENTO CONSTRUCTIVO BOTTOM-LEFT (BLP)

El procedimiento Bottom-Left (BLP) es un método sencillo, conocido para asignar un conjunto de piezas en un rectángulo de stock. Este método consiste en colocar las piezas en el rectángulo siguiendo un orden preestablecido.

La primera pieza se coloca en la esquina inferior izquierda del rectángulo. En cada iteración, la siguiente pieza en el pedido se selecciona y se mueve hacia abajo y hacia la izquierda tanto como sea posible. Luego, para cada ordenación inicial de las piezas, el algoritmo produce una solución. Proponemos ahora algunas variantes del BLP para usarlas como base de comparación en nuestros experimentos, así como para formar parte de métodos más complejos.



Se muestra un rectángulo estándar S (en el lado izquierdo) en el que queremos cortar las dos piezas que se muestran debajo, se muestra una tira sin recortar (en el centro de la figura) y otra con recorte (en el lado derecho). Esta imagen muestra claramente que si no se permite el recorte podemos obtener una cantidad relativamente grande de residuos en la tira; mientras que el recorte permite encajar piezas más pequeñas para completar la tira de forma más eficiente.

En nuestra implementación, ordenamos las piezas por ancho no creciente wi en la primera iteración, seleccionamos la pieza con mayor w-valor, disminuir su demanda b por una unidad y asignarlo por BLP, colocando en la esquina inferior izquierda del rectángulo.

Teniendo en cuenta que, considerando la restricción de 2 etapas de nuestro problema, el ancho de esta primera pieza define el ancho de la primera tira. En la segunda iteración, seleccionamos la primera pieza del pedido con demanda positiva, disminuimos su demanda en una unidad y la ubicamos en la primera franja, junto a la pieza previamente asignada. Tenga en cuenta que el ancho de la pieza actual es inferior o igual al ancho de la pieza de la iteración anterior; por lo tanto, es menor o igual al ancho de la tira.

Seguimos así hasta el largo de la pieza seleccionada excede la longitud restante de las tiras (la diferencia entre L y la suma de las longitudes de las piezas añadidas en iteraciones anteriores a esta tira). A continuación, la nueva pieza se asigna en una nueva franja. El algoritmo termina cuando se han satisfecho todas las demandas o no hay espacio para una nueva tira.

La figura 2 muestra un ejemplo en el que el BLP está construyendo una solución. En la primera iteración, se seleccionó y cortó la pieza 3 (esquina inferior izquierda) Si consideramos que $b_3 = 2$,luego en la segunda iteración, seleccionamos la segunda unidad de la pieza 3 y la colocamos en la misma tira, al lado de la unidad anterior.

Suponga que la siguiente pieza en el pedido es la número 7 y $_7 = 3$, por lo que en la iteración 3, se asigna una unidad de esta pieza en la tira 1. En la iteración 4, se selecciona

la segunda unidad de la pieza 7, pero no hay espacio para ella en la primera tira, por lo que asignamos esta unidad en una segunda tira. Seguiríamos así con el resto de piezas.

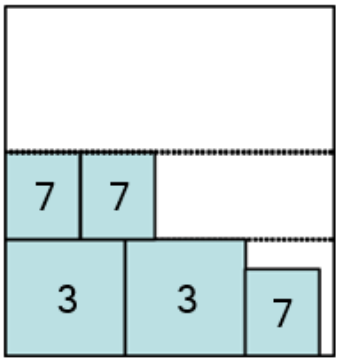


Figura 2. Ejemplo de BLP

Cabe señalar que el BLP inicia con una nueva tira cada vez que la siguiente pieza en el pedido no cabe en la tira actual (su longitud es mayor que la longitud de la tira restante). Sin embargo, otra pieza i una posición posterior en el pedido podría caber en la tira actual $l_i \leq r_s$.

A continuación, antes de comenzar una nueva tira, recurriremos al procedimiento REMAIN para llenar este vacío. Este procedimiento busca la primera pieza i en el pedido con demanda positiva $\mathrm{b}_i, satisfactorio\, \mathrm{l}_i \leq r_s$ lo asigna en la franja y después b_i y r_s actualizan y el método continúa llenando el espacio restante hasta que no queda ninguna pieza i satisfactorio $l_i \ \leq r_s$ encontrado.

En la Figura 2 aparece un pseudocódigo del BLP para producir una solución al problema. El pseudocódigo está escrito usando notación matemática general. Sin embargo, la implementación real aprovecha estructuras de datos eficientes y mecanismos de actualización rápida que son difíciles de representar matemáticamente.

También implementa una regla de desempate para la selección de piezas que se basa en la relación costo/superficie. La regla se utiliza cuando más de una pieza tiene el mismo ancho y existe un empate en el orden de las piezas, luego se resuelve el empate según la proporción, entrando primero la pieza con mayor proporción.

Inicialización:

Ordena las piezas según w_i,resuelve los empates según $c_i/\ (l_i w_i)$

Cree la pieza tira en la parte inferior del rectángulo de stock, añade la primera pieza i del pedido a la tira $b_i = b_i\ - 1$

Sea $Wlibre$ la anchura del rectángulo sin banda: $W\ libre\ = W\ - w_i$

$r_s\ =\ L\ -\ l_i$

Mientras exista una pieza i con $b_i > 0$

Sea i la siguiente pieza en la ordenación con $b_i > 0$ Si $(l_i \leq r_s)$

Añadir la pieza i a la tira $b_i = b_i - 1$

Valor = valor + c_i

$r_s = r_s - l_i$

En caso contrario Llamar al procedimiento REIMAN

Sea i la primera pieza de la ordenación con $b_i > 0$ y $w_i\ \ libre$

Si tal pieza i existe Crear nueva tira

$Wlibre\ =\ Wlibre\ + w_i$

Añade la pieza i para comenzar la tira $b_i = b_i - 1$

Valor = valor + c_i , $\quad r_s = r_s - l_i$

En caso contrario Stop

Procedimiento de REMAIN

$R =\ \{pieza\ j/b_j > 0,\ l_j \leq r_s\}$

Mientras $(R = \emptyset)$

Seleccione la pieza i en la ordenación $/i$

Añadir i a la tira $b_i\ = b_i - 1$, $r_s = r_s - l_i$

Valor = valor $+c_i$

$R\ =\ R - \{i\}$

El método BLP se basa en que el ordenamiento inicial de las piezas, según w, garantiza que la pieza seleccionada en una iteración pueda ubicarse dentro de la franja creada con una pieza previamente seleccionada. Sin embargo, con una pequeña modificación, podemos adaptar este método para construir una solución a partir de cualquier orden inicial.

Solo necesitamos reemplazar la condición" $(l_i \leq r_s)$ " en el exterior Si la declaración con la condición" $(l_i \leq r_s y\ w_i \leq W\ banda)$ ", dónde tira es el ancho definido como el ancho de la primera pieza de la tira. Ahora, cuando seleccionamos la siguiente pieza en el pedido, si no se puede ubicar en la franja actual en construcción (por su largo o por su ancho), comenzamos una nueva franja con esta pieza, pero primero recurrimos a la rutina RESTANTE para completar la franja actual. Nos referiremos a este método, BLP general como el GBLP.

CAPITULO 4: ALGORITMOS METAHEURDITICOS

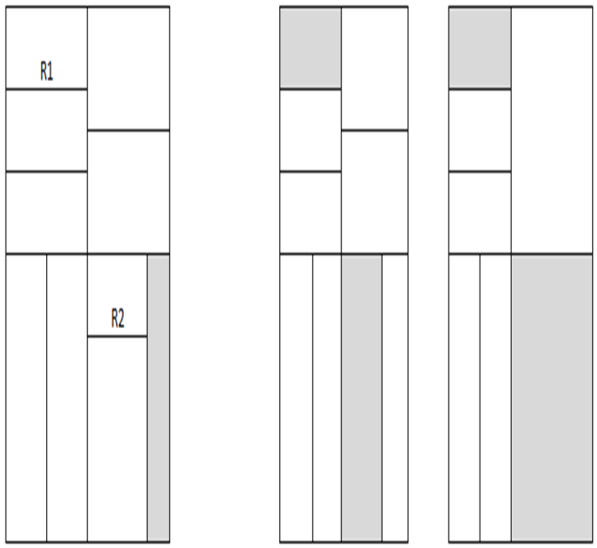
Tabu Search Es actualmente un metaheurístico de optimización ya establecido para una introducción Ver Glover y Laguna (1997) [60] los elementos básicos del algoritmo se describen cómo sigue.

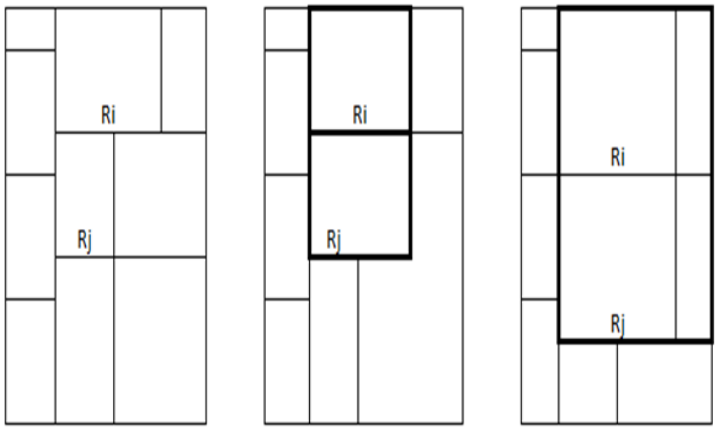
4.1 Definición De Movimiento

La primera etapa al diseñar un procedimiento Tabu Search Es la definición de un movimiento que mantenga la propiedad de corte de guillotina. Se dice que dos rectángulos son vecinos si comparten un lado parcialmente. Puede ser un lado completo correspondiendo a la definición de adyacente de la sección anterior, solo una parte o únicamente un vértice.

Se define el movimiento de la siguiente forma. Tomamos una pareja de rectángulo vecino y definimos la región común como el menor rectángulo que contiene ambos. Después determinamos los cortes de guillotina más cercano a la región que la contienen.

El nuevo rectángulo definido por este corte es vaciado y se corta nuevamente siguiendo el procedimiento de GRASP de la sección de esta manera, En vez de cortarlo siguiendo el procedimiento constructivo. Ejecutamos varias iteraciones de la fase constructiva y de la mejora del GRASP.





4.1.1 Selección Del Movimiento

En cada iteración seleccionamos el movimiento a realizar siguiendo 3 etapa:

• Primero se selecciona un rectángulo de forma aleatoria de la lista de todos los rectángulos, pieza y desperdicios, de la solución actual.

• Segundo consideramos todos los rectángulos que son sus vecinos, de uno en uno.

• Finalmente, para cada parte rectángulos, seguimos la etapa explicada arriba en la definición anterior de movimiento para obtener un nuevo rectángulo acortar.

Sí este rectángulo no contiene desperdicio, no se considera su corte, ya que es imposible mejorar la solución actual en problemas sin peso y altamente improbable en los con pesos. Si el rectángulo contiene algún desperdicio, espaciado y cortado nuevamente, en este contexto cada iteración de un procedimiento GRASP puede verse como una exploración de posibles movimientos diferentes, debido a que cada iteración GRASP puede producir al final un patrón de corte diferente, el movimiento a realizar es seleccionado como él como el que produce el incremento máximo en el valor de la solución actual.

4.1.2. Lista Tabu

Para cada movimiento se guardan en la lista Tabu las dimensiones y posición del nuevo rectángulo que se ha cortado (coordenada del extremo inferior izquierdo y superior derecho) y el tipo de pieza cortada en esquina inferior izquierda. Por lo tanto, en la siguiente iteración; se puede considerar cortar el mismo rectángulo, pero no puede usarse el mismo patrón.

La longitud de la lista cambia dinámicamente, después de un número dado de iteraciones sí mejorar la mejor solución conocida. Sí mes la raíz cuadrada del número de piezas, la longitud se selecciona aleatoriamente en el intervalo. $[0.5m^*,\ 1.5m^*]$.

4.1.3. procedimiento Básico Tabu Search

Sea S la solución actual con un conjunto asociado de rectángulo R, un conjunto de (piezas y desperdició) Un conjunto de Cortés guillotina C valor v_s. Sea S la mejor solución conocida con valor v.

Inicialización

Sea $S^* = S = SI$, la solución inicial obtenida usando el algoritmo de GRASP

Sea $niter\ = 0$

Iteración: Mientras $(niter < maxiter)$

Etapa1.

Seleccionar aleatoriamente un rectángulo $R_i \in R$

Sea $S_i = \emptyset$ la solución parcial construida a a partir de R_i y $v_i = 0$ es el valor de la solución completa que contiene S_i

Etapa2.

Sea $N_i = \{R_j \in R | R_j \; donde \; R_i\}$ es vecino de R_i

Para cada, $R_j \in N_i$:

1. Construir $R_{i\ j}$, el menor rectángulo que contiene a R_i *y* R_j

2. Determinar Pij, el menor rectángulo tal que:

a. $R_{ij} \subset P_{ij}$

b. Cada lado de P_{ij} esta incluido en un corte de C

3. Si P_{ij} no contiene desperdicio regresar y tomar el siguiente R_{ij}

4. Para cada rectángulo $R_k \in R$ tal que $R_k \subset P_{ij}$

a. $R = R - \{R_K\}$

b. $v_s = v_s - v_k$

5. Para $n = 1$ hasta max

Resolver el problema de corte correspondiente al rectángulo P_{ij} usando el algoritmo de GRASP.

Sea v_s el valor de la solución de S_n obtenida.

Si el movimiento no es Tabú y $v_n + v_s > v_i$ *ó* $v_n + v_s > v^*$ criterio de aspiración.

i. $v_i = v_s + v_n$

ii. $S_i = S_n$

iii. $P_i^* = P_{ij}$

6. Recuperar el rectángulo original R *y* v_s y tomar el siguiente $R_j \in N_i$

Etapa3.

Actualizar la solución actual de S sustituyendo los rectángulos de P_i^* por todos los asociados a S_i y $v_s = v_i$

Si $v_i > v^*$ actualizar la mejor solución conocida S^*

Actualizar la lista tabú
$niter = niter\ + 1$;

4.1.4 Estrategia de intensificación y diversificación

La definición del movimiento involucra el procedimiento de búsqueda un importante grado de diversificación.Si el rectángulo seleccionado está situado próximo a la esquina superior derecha, puede producir un cambio local, pero está situado cercano a la esquina inferior izquierda, entonces es posible que se corte nuevamente la mayor parte del rectángulo inicial. Por lo tanto, el movimiento puede cambiar drásticamente la estructura de la solución. La

selección aleatoria del primer rectángulo considerado el movimiento que asegura que a largo plazo tendremos algunos movimientos en los que la solución cambiar a casi totalmente.
No obstante, el procedimiento de evaluación No cambia y se basa en una corta unidimensional

que puede llevar a decisiones erróneas. Por lo tanto, Hemos diseñado un cambio en la función de evaluación qué puede ser usado con el objetivo de intensificar y diversificar.
En el proceso de búsqueda, se guarda un conjunto de n solución de élite, esto es el conjunto

de la n mejores soluciones obtenida hasta el momento. Sí f_{ik} es el número de veces que la pieza i aparece en la solución k, $y\ M$ es el número total de piezas contenidas en la n solución, podemos definir un valor modificado de la pieza i.
$P_i = v_i\,(1 + \beta\,(\sum_k f_{ik}/M))$
Si el parámetro$\beta > 0$, entonces el valor de las piezas que aparecen más frecuentemente la

solución Se incrementa y su presencia es favorecida, por lo contrario,$\beta < 0$, si la pieza con alta frecuencia tendrá menor valor y la otra pieza tiene más posibilidades de aparecer en la solución, en una estrategia de diversificación. Estos valores modificados se usan en el cálculo de las cotas $BK_1\ y\ BK_2$ procedimiento de decisión para seleccionar una pieza a cortar.

4.2 PATH RELINKING

Si se tiene un conjunto de soluciones de referencia, qué es típicamente un conjunto de soluciones de gran calidad, entonces Path Relinking generan nuevas soluciones explorando los caminos que las conectan comenzando desde las soluciones llamadas soluciones iniciales, Una serie de movimientos que definen un camino en el espacio de soluciones que llevan a otras soluciones llamadas soluciones guías.

Esta técnica es llamada Path Relinking ya que Tabu Search en dos soluciones cualesquiera se enlazan por una serie de movimientos que se ejecutan durante la búsqueda Ahora estás soluciones se unen otra vez, pero siguiendo un camino diferente en el cual los movimientos no son seleccionados según el mejor valor de la función objetivo, sino buscando movimientos que lleven directamente a la solución guía.

Aunque puede usarse siempre que esté disponible un conjunto de referencia.Con nuestro caso, aplicamos Path Relinking a un conjunto de soluciones de gran calidad obtenida por el algoritmo Constructivo y el algoritmo Tabu Search. Iniciamos desde un caso extremo de una solución inicial, una solución de valor cero en la cual se tiene el rectángulo original sin cortar. Tomamos como solución guía una de las soluciones de élite obtenida previamente con cualquiera de los algoritmos. Seguimos una estrategia constructiva introduciendo uno por uno los cortes de guillotina definidos por la solución guía.

Cada vez que se añade un nuevo corte de guillotina por solución guía, Se corta un rectángulo de la solución actual y aparece dos nuevos rectángulos. Al final del proceso repetimos la solución guía, pero entre tanto en cada etapa tomamos la lista de rectángulos actual y aplicamos el algoritmo Constructivo para cortar cada uno de ellos y obtener una solución completa, en la cual se fuerza parte de la estructura de corte de la solución guía.

4.2.1 Procedimiento Path Relinking

Sea S el conjunto de soluciones referencias

Sea S^* la mejor solución conocida con valor v^* y la lista de piezas $P*$

Denotamos por R_0 el rectángulo inicial a cortar

Para cada $S \in S$, repetir el siguiente procedimiento.

Inicialización:

Sea C la lista ordenada de corte guillotina de S

Sea $L = \{R_0\}$ la lista de rectángulos aun por cortar

Sea $P = \emptyset$ el conjunto de piezas ya cortadas

Sea $v = 0$ el valor de la solución construida.

Etapa1.

Tomar el siguiente corte $c \in C$

Este corte divide algún rectángulo $R_i \in L$ en dos nuevos rectángulos $R_i^{,}, R"i$

Sea $L = L \cup \{R_i^{,}\}" - Ri$

Eliminamos de P las piezas incluidas en R_i

Sea $v = v - \sum_{j \epsilon R_i} v_j$

Etapa2.

Resolver los problemas de corte de los rectángulos $R_i^{,}$ y $Ri"$ con el GRASP

obteniendo las piezas $P_i^{,}$ y $Pi"$ y valores $v_i^{,}$ y $vi"$.

Hacer $P = P \cup P_i^{,}"$

Hacer $v = v + v_i^{,} + vi"$

Etapa3.

Si $v > v^*$ Actualizar S^*

Si se han tomado todos los cortes en C ir a la siguiente solución referencia $S \in S$ En caso contrario, regresar a Etapa1.

Ejemplo:

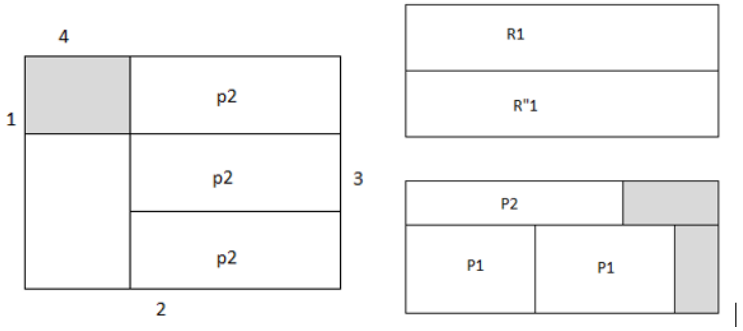
Consideremos el ejemplo, tomado de viswanathan y Bagchi [114]

Con $L = 5,\ w = 3$ y $m = 2$

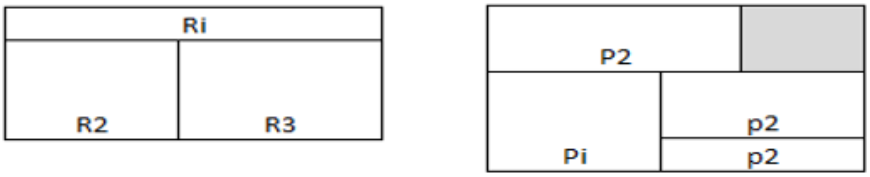
P_i	l_i	w_i	v_i	b_i
P_1	2	2	25	2
p_2	3	1	10	5

Supongamos que tenemos una solución de referencia en la figura en la cual hemos cortado

una pieza de tipo 1 y tres piezas de tipo 2 con un valor total de 55. La secuencia de corte de guillotina está marcada fuera del rectángulo. Iniciamos el proceso de construir una nueva solución considerando el rectángulo inicial $R_0 = (L, W)$ y aplicamos el primer corte de la solución referencia, obteniendo en la figura las dos regiones$R1"yR2"$ después cortamos está dos regiones usando GRASP, obteniendo la solución de la figura.



Solución referencia en el procedimiento Path Relinking.



Luego imponemos el segundo corte de la solución de referencia y obtenemos la figura en la cual ha sido divididos en $R_{2\cdot}$, $y\ R2"$, $R1$, ya fue resuelto y ahora resolvemos para $R_{1'}\ Y\ R2"$ obteniendo la figura. Procediendo de la misma manera imponemos los cortes 3 y 4 la solución final será igual a la solución de referencia, pero en el proceso de construcción hemos hallado una mejor solución.

4.3 GENERACIÓN DE COLUMNAS

La primera idea sobre generación de columna fue desarrollada por Ford y Fulkerson (1958) [51] para resolver. Problemas de flujo de redes, después esta idea fue generalizada por Dantzing y Wolfe (1960) [26] Gilmore y Gomory también aplicaron el método generación de columna al problema de corte en una dimensión. Pero solamente durante los últimos 10 años este procedimiento ha sido explotado computacionalmente con gran éxito. Una introducción

reciente del método de generación de columna para programación entera ha sido presentada por Barnahart et al. (1994) [6].

El método de generación de columna has pasado ser un procedimiento eficiente para

diferentes problemas de optimización. tales como la Ruta de vehículos y secuenciación. Un amplio Survey sobre el método dado por Desrosiers et al (1995) [29]. Generalmente, cuando un problema lineal contiene demasiadas variables para ser resuelto explícitamente, se inicializa con un conjunto de variables y se calcula una solución óptima de este subproblema. Después se verifica si añadir una variable, que no está en la formulación lineal actual, Puede mejorar la solución lineal. De acuerdo a la teoría de programación matemática, esto se puede hacer calculando los costes reducidos de la variable.

En un problema lineal de la forma $max\,\{e^t\,x|\,Ax \leq b_i\ ,\ x \geq 0\}$ una variable que tiene coste reducido positivo, puede mejorar la solución. El coste reducido r_j de una variable no básica j Con columna asociada $a_j \in R^m$ y coeficiente de coste c_j $correspondienteaunasoluciónlinealconvariabledua$ ϵR^m si se define como $r_j = c_j - y^t a_j$ si no hay variable que tenga coste, reducido positivo, entonces, la solución actual es óptima y el problema original ha sido resuelto.

4.3.1 Algoritmo de Generación de columnas

1. Seleccionar un pequeño subconjunto de variable $J = \{1, \ldots n\}$

2. Obtener una solución óptima básica $\bar{x_j}$ del problema lineal ${e^t}_j\bar{x_j} = max\ \left\{{e_j}^t\ x|\,A_j x_j \leq b, x_j \geq 0, x_j \in\right.$ y determinar los valares de las variables duales $\bar{y}$
3. Si $r_i = c_i - y^{-t}a_i \leq 0\ \forall i\epsilon\,\{1, \ldots n\}$ entonces parar
4. Añadir columnas j con $r_j > 0\ a\ j$
5. Regresar a pasos

El problema ${e^t}_j\bar{x_j} = max\ \{{e_j}^t\ x|\,A_j x_j \leq b, x_j \geq 0\}$ es llamado problema master mientras el problema lineal ${e^t}_j\bar{x_j} = max\ \left\{{e_j}^t\ x|\,A_j x_j \leq b, x_j \geq 0, x_j \in R^{|j|}\right\}$ que contiene un subconjunto de columnas del problema *master* restringido.

4.3.2 Método de Generación de Columna

Para el problema de corte en dos dimensiones considerado aquí Gilmore y Gomory Propusieron

resolver el siguiente programa de entero.

$$min \ \sum_{q\in Q} c_q x_q$$
$$s.a \ \sum_{q\in Q} c_{iq} x_q \geq d_i, \quad i = 1, \ldots, m$$

$x_q \geq 0$ y enteras, $\forall q \in Q$

Donde Q es el conjunto de todos los posibles patrones de corte de dos dimensiones para todo tablero R_p, x_q el número de veces que el patrón q es usado solución aiq es el numero de veces de pieza i aparece en el patrón q, d_i la demanda de la pieza i y c_q el coste del patrón q.

Usualmente este coste corresponde al coste del tablero R_p ,así que podemos denotar por c_p. Si el objetivo es minimizar entonces $c_p = L_p W_p$ Si, el número de tablero rectangular de tipo p está acotado por N_p el modelo, tiene el siguiente tipo de restricciones.

$$\sum_{q\in Q_r} x_q N_p, \leq \forall p$$

Donde Q_p es el conjunto de patrones usando tablero rectangular. Sin embargo, en la mayoría de las situaciones de la vida real, los tableros requeridos están disponibles o pueden ser fácilmente obtenidos, y estas restricciones no es requerido. Como el conjunto Q de patrones no puede ser descrito completamente, excepto para los problemas, muy pequeño, entonces Gilmore y Gomory desarrollaron un procedimiento de generación de columna para el problema de corte que puede resumirse de la siguiente manera.

1. Generar el conjunto inicial Q de m patrones de corte, donde cada patrón contiene un tipo de pieza.

2.Resolver la relajación lineal del problema formulando arriba considerando solamente las variables correspondientes a los patrones en Q

3. Para cada tipo de tablero R_p resolver los subproblemas.

$$Z_p = \ Max \sum_i \pi_i a_i$$
$$s.a = \{a_{1.....a_m}\} \text{ es el posible patrón de corte para } R_p$$

Dónde π es el vector de precios duales de la solución de problemas lineal. Si para algún valor de p, $z_p > c_p$ eentonces la columna correspondiente a esta solución es agregada a Q y se regresa a la etapa 2 para resolver problema lineal aumentado, de lo contrario, la solución actual se redondea para obtener una solución entera y termina el procedimiento.

Ahora la pregunta es, cómo resolver de forma eficiente en su problema de la etapa 3 Para que describimos en la siguiente sección, método exacto y aproximado. Hemos de hacer notar que aún sí resolvemos el subproblema por método exacto, el procedimiento total heurístico, debido a la etapa de redondeo. La calidad de la solución entera dependerá no solamente del algoritmo usando para resolver su problema, sino también del procedimiento de redondeo y de la estructura de la demanda.

CONCLUSIONES Y NUEVAS LINEAS DE INVESTIGACIÓN

1. La revisión del estado del arte nos permite nos permite comprender y describir los últimos avances en el desarrollo de algoritmos eficientes para el problema de corte en dos dimensiones.

2. El desarrollo metodológico de algoritmos heurísticos constructivos basados en la colocación de piezas con orientación fija, rotada y corte en franjas, nos permiten obtener patrones de corte de buena calidad

3. El desarrollo metodológico de algoritmos Metaheurísticos basados en las soluciones obtenidas con los algoritmos constructivos, nos permiten mejorar estas soluciones minimizando el desperdicio.

El tratamiento metodológico de estos algoritmos me motiva a continuar profundizando en el problema TDC a nivel teórico, así como realizar la implementación computacional de los mismos, para intentar mejorar los resultados publicados por otros autores.

BIBLIOGRAFIA

1. Alvarez-Valdes, R., A. Parajon, J. M. Tamarit. 2002. A tabu search algorithm for large-scale guillotine (un)constrained two- dimensional cutting problems. Comput. Oper. Res. 29 925–947.

2. Beasley, J. E. 1985. Algorithms for unconstrained two-dimensional guillotine cutting. J. Oper. Res. Soc. 36 297–306.

3. Belov, G., G. Scheithauer. 2003. A branch-and-cut-and-price algo- rithm for one-dimensional stock cutting and two-dimensional two-stage cutting. Technical Report MATH-NM-03, Institute of Numerical Mathematics, Dresden University, Dresden, Germany.

4. Resende, M. G. C., C. C. Ribeiro. 2001. Greedy randomized adaptive search procedures. F. Glover, G. Kochenberger, eds. State-of-the- Art Handbook in Metaheuristics. Kluwer Academic Publishers, Boston, MA, 219–250.

5. Sweeney, P. E., E. R. Paternoster. 1992. Cutting and packing prob- lems: A categorized applications-oriented research bibliogra- phy. J. Oper. Res. Soc. 43 691–706.

6. Christofides, N., Whitlock A. (1977). An algorithm for twodimensional cutting problems. Op- erational Research. Vol. 25. pp. 30-44.

7. Dyckhoff, H. (1990). A Typology of Cutting and Packing Problems. European Journal of Oper- ation Research 44, 145-159.

8. Farley, A. (1990). The cutting stock problem in the canvas industry. European Journal of Oper- ational Research, 44, 247-255.

9. Gilmore, P., Gomory, R. (1961). A linear programming approach to the cutting stock problem. Operation Research. 9 849-859.

10. Kantorovich, L., Zalgaller V (1951). Calculation of Rational Cutting of Stock. Lenizdat, Leningrad. 11. Kantorovich, L. (1960). Mathematical Methods of Organizing and Planning Production. Man- agement Science 6, No. 4. 363-422.

12. Martello, S., Toth, P. (1990). Knapsack Problems. Great Britain: John Wiley Sons.

13. Mathur, K., Solow D. (1996). Investigación de Operaciones. México: Prentice-Hall His- panoamericana S.A.

14. Mobasher, A, Ekici, A. (2012). Solution approaches for the cutting stock problem with setup cost. Computers Operations Research.

15. Oliveira, J., Ferreira, J. (1990). An improved version an improved version of Wang´s algorithm for two - dimensional Cutting Problems. EJOR 44. pp. 256-266.