Problema de corte en una y dos dimensiones.

Alex Escárate, Alonso Sandoval, Hernán Vargas.
Universidad Técnica Federico Santa María
Avenida España 1680, Valparaíso, Chile
alex.escarate@alumnos.usm.cl, alonso.sandovala@alumnos.usm.cl, hernan.vargas@alumnos.usm.cl.

***Resumen*—El presente documento plantea las características básicas y resolución por programación lineal del llamado problema de corte, tanto en una como dos dimensiones. Establece la formulación matemática y las consideraciones a tener en cuenta para la resolución del problema, además ejemplifica el corte en dos dimensiones para el rubro de la creación de muebles y extiende dicha problemática con límites y costos variables en la utilización de los patrones.**

***Palabras Claves*—Problema de corte, Problema de la guillotina, Programación lineal, Corte en una dimensión, Corte en dos dimensiones, Patrones de corte fijos, Patrones de corte con costo, Patrones limitados.**

I. INTRODUCCIÓN

El problema de corte, más conocido como "The cutting stock problem", es un problema clásico de optimización que surge de la necesidad de la industria de mano-factura de mejorar sus procesos productivos. Comúnmente lo que se busca es minimizar el uso de materia prima para la generación de productos más pequeños, por ejemplo, una plancha de metal de área $A \times L$ para la producción de diferentes planchas de menor tamaño con áreas: $a_1 \times l_1$, $a_2 \times l_2$, etc.

Este problema ha sido estudiado por personas desde el área de la programación hasta gente de negocios, en donde diversas metodologías se han desarrollado en la búsqueda de la optimización de los recursos disponibles. Para las empresas actuales, el desarrollo de estos métodos significa un importante impacto en la disminución de costos, lo que las vuelve más competitivas, factor fundamental en el sistema económico moderno.

El problema en sí tiene muchas variantes, todas dependiendo del contexto de la industria donde se aplica. La formulación más general consta de un "stock" disponible de materiales a cortar para la producción de pedidos más pequeños como mencionábamos anteriormente. La forma en que cortamos la materia prima la denominamos "patrones", las cuales determinan las unidades de cada pedido a generar a partir del material del "stock".

La función objetivo varía dependiendo de las necesidades de la empresa. La más común es minimizar los residuos que se generan de los patrones de corte dado que estos no son perfectos. Otro enfoque podría ser minimizar los costos asociados a la maquinaria que genera los patrones de cortes, es decir, considerar el coste de generar un patrón como variable, así se requeriría elegir la combinación de actividades (que realizan el corte de acuerdo a un patrón) más económica. También se pueden considerar las dimensiones de la materia prima a cortar, considerando sólo el largo(1D), el largo y el ancho(2D), e incluso volúmenes(3D). Acorde a este último criterio el problema adquiere mayor complejidad a medida que aumentamos las dimensiones.

Los artículos más utilizados en la investigación y construcción de este documento son los escritos de Gilmore y Gomory[4][6], la tesis presentada por Rosa Delgadillo en el 2007[2] y el paper de Tzu-Yi Yu et al. del 2014[1].

A continuación realizaremos un barrido histórico por las diferentes soluciones al problema, planteando al final el modelo matemático que contemplaremos en este proyecto.

II. ESTADO DEL ARTE

El problema del corte es `NP-Completo` lo que además de implicar que no existen algoritmos que puedan resolver el problema en tiempo polinomial, tiene como consecuencia que podamos hacer la reducción de éste a uno de los 21 problemas de Karp[3]: El problema de la mochila, razón por la cual, toda investigación de dicho problema también nos es útil a la hora de analizar el problema de corte, es más, podemos tratar indistintamente tanto los problemas de la mochila (corte en 1D), los de corte (2D) y los de almacenamiento o empaquetamiento (corte en 3D) que afectan constantemente a la industria, puesto que sus soluciones serán prácticamente las mismas.

Debido a los factores mencionados anteriormente, el problema del corte ha sido estudiado y documentado ampliamente desde los años 60. Los escritos más importantes de dicha época son sin duda los de P.C Gilmore y R.E Gomory que en 1961 publican una resolución del problema de corte en una dimensión por métodos de programación lineal, además nos dan una descripción general de la situación en dos dimensiones y el problema de la guillotina[4]. En 1963 extienden su obra y la contextualizan en la industria del papel[5] y en 1965 amplían su solución analizando los problemas en dos o más dimensiones como una colección de problemas unidimensionales[6] reducibles al problema de la mochila cuya solución ya habían descrito.

Después de sus trabajos el problema de corte en dos dimensiones ha sido contextualizado a diferentes industrias con necesidades específicas. En general, el problema puede ser clasificado dependiendo de sus patrones de corte, así podemos desglosar en problemas con guillotina o sin ella. El problema de la guillotina radica en que en ciertos rubros para minimizar los gastos es necesario que los cortes a efectuar sean ininterrumpidos y rectos desde un lado de la placa hasta el otro, luego el problema de la guillotina queda definido por el número mínimo de cortes necesarios para crear un patrón, la figura 1 es un ejemplo de ello.

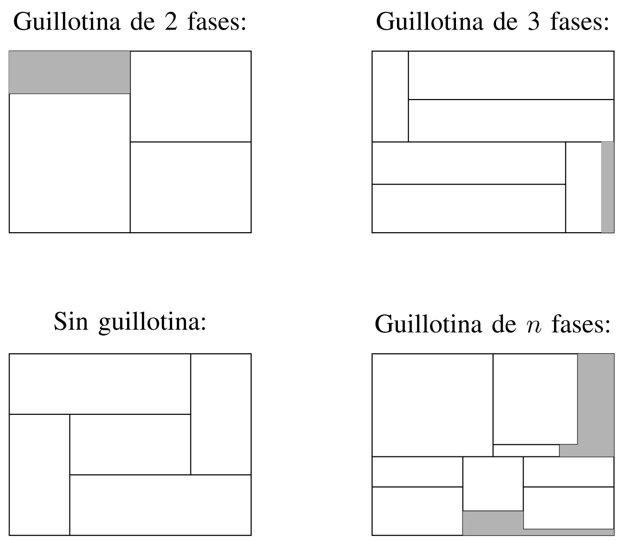

Figura 1. Ejemplo de patrones para cortes con guillotina.

En 1985 Beasley presenta un modelo de programación dinámica que, por medio de la recursión, logra obtener múltiples soluciones óptimas para problemas de corte por guillotina[7], aunque solo es posible en pequeña escala. Evitando los problemas de la recursión, Christofides y Whitlock[8] en 1977 solucionan el mismo problema con arboles de búsqueda, método que mejorará su eficiencia por el trabajo de Viswanathan y Bagchi en 1993[9] y el de Hifi y Ouafi en 1997[10] que, aplicando Best-First, empeoran las soluciones, pero mejoran los tiempos.

Para la creación de patrones, en 1983 Wang presenta dos métodos combinacionales[11] los cuales son mejorados posteriormente por Vasko[12] en 1989 y por Oliveria y Ferreira[13] en 1990. Dichos métodos aún son ampliamente utilizados.

El 2000, Valdes et al.[14] presentan un algoritmo de búsqueda tabú[15] para solucionar problemas de corte generales. Posteriormente utilizan un algoritmo GRASP[16] para mejorar sus soluciones óptimas y disminuir la perdida de material en patrones con corte de guillotina de más de dos fases. Si bien, este último algoritmo muestra un mejor desempeño también requiere mayor tiempo computacional.

Tanto los algoritmos de programación lineal como los dinámicos deben buscar entre todas las combinaciones la mejor solución posible, debido a la carga computacional que supone dicha tarea, es de esperar que la mejora constante que tienen los procesadores ayude significativamente en la resolución. Actualmente es posible obtener soluciones en tiempos aceptables para una gran cantidad de situaciones con los algoritmos descritos anteriormente, pero, cuando la cantidad de patrones posibles incrementa, el tiempo computacional también lo hace, y no polinomialmente, por ello algunos algoritmos sacrifican la exactitud de sus soluciones en busca de tiempos más plausibles para la industria.

En este contexto los algoritmos genéticos[17], que utilizan la noción de la supervivencia del más fuerte y los algoritmos de recorrido simulado[18][19], que deciden la solución óptima basados en probabilidades, han sido los pioneros en la investigación actual del problema del corte.

Un ejemplo de algoritmo de recorrido simulado para la creación de patrones sin guillotina es el de Lai y Chan[20]. Faina utiliza el mismo tipo de algoritmo pero quita la restricción de guillotina en su trabajo de 1999[21]. Una variante de solución utilizando arboles binarios en conjunto a un algoritmo de camino simulado fue propuesto por Parada et al. en 1996[22]. Por otro lado, los algoritmos genéticos no se quedan atrás en la investigación, varios autores enfocan su trabajo buscando soluciones en ellos como por ejemplo: Jakobs en 1993[23], Wu en 2006[24] o Goncalves en 2007[25], entre otros.

En general, la estructura de un problema de corte puede ser declarada según Dyckhoff[26], de la siguiente manera:

1. Existen dos grupos de parámetros básicos, cuyos elementos son cuerpos geométricos en una o más direcciones:
   - Un conjunto de órdenes que determinan cierta cantidad de piezas con tamaños, formas y cantidades específicas.
   - Un stock grande de bloques de material de tamaño fijo del cual deben cortarse las piezas determinadas por la demanda.
2. Existe un proceso que determina los patrones por los cuales se pueden cortar los bloques de material de manera de obtener una o más piezas demandadas. Puede cumplir alguna de las siguientes condiciones:
   - Especificaciones del material a cortar.
   - Tecnología de corte utilizada.
   - Existencia de objetivos específicos a alcanzar.

Dependiendo de las características de la producción, el proceso de creación de patrones es, en general, mucho más complejo que la asignación óptima de ellos. Al utilizarse algoritmos combinatoriales sujetos a las condiciones de la maquinaria el problema de hallar todos los patrones de corte posibles requiere mucho tiempo computacional. En el modelo matemático dejaremos de lado la creación de patrones y nos enfocaremos directamente en la resolución del problema del corte, es decir: la optimización de productos generados sujetos a los a un stock y patrones de corte suministrados.

III. Modelo Matemático o LP

El enfoque principal de la literatura al cutting stock problem es para los casos de 1 y 2 dimensiones, siendo el caso 1D el punto de partida. Es por esto que en primer lugar explicaremos el modelo matemático para el caso de una dimensión, generalizando con el modelo para dos dimensiones. La siguiente formulación está basada en el estudio algorítmico del problema de corte, realizado en la tesis de Delgadillo [2]

III-A. *Formulación Estándar, una dimensión*

Supongamos que tenemos un stock de barras de largo L a cortar en trozos más pequeños de largo l_i, para satisfacer ciertas demandas N_i. Consideremos además que tenemos un conjunto de patrones j, con $j = 1, 2, ...$, es decir, el j-ésimo patrón representa una forma de cortar un trozo de barra de largo L para formar barras más pequeñas de largo l_i. Luego, nuestras variables serán los x_j que representan la cantidad de veces que utilizaremos el patrón j.

Así, tenemos la formulación más simple para el caso de una dimensión:

$$\begin{aligned} \text{Min } & \sum_j x_j \\ \text{s.a. } & \sum_j a_{ij} x_j \geq N_i, \text{ i=1,...,m} \\ & x_j \geq 0 \end{aligned}$$

Gilmore y Gomory[4], expresaron este modelo de forma matricial quedando:

$$\begin{aligned} \text{Min } & \vec{x} \\ \text{s.a. } & A\vec{x} \geq \vec{n} \\ & \vec{x} \geq 0 \end{aligned}$$

Donde:

- $\vec{x}$ representa un vector columna de las variables x_j, que representan la cantidad de veces que se utiliza el patrón j.
- A representa la matriz de m filas, donde cada columna $[a_1, a_2, ..., a_m]^T$, representa un patrón de corte. El elemento a_1, por ejemplo, representa la cantidad de elementos de largo l_1 que se generan.
- $\vec{n}$ representa el vector columna de las demandas N_i

Podemos notar que el modelo en una dimensión puede ser reducido al planteamiento del problema de la mochila con la siguiente idea: Supongamos que nuestras barras de largo L se transforman en una mochila que soporta un peso P, luego las secciones que podemos cortar de la barra (de largo l_i) serán equivalentes a los objetos, de peso p_i, que podemos llevar en nuestra mochila. El problema de corte además incluye "mochilas infinitasz su solución implica la minimización de estas para transportar toda nuestra demanda y nuestros patrones serán las configuraciones utilizadas en dicha misión.

III-B. Formulación Estándar, dos dimensiones

Para el caso 2D consideremos un stock de láminas de área $L \times W$, para formar otras de menor tamaño de área $l_i \times w_i$ (las cuales llamaremos piezas) con el fin de satisfacer cierta demanda de N_i.

Como dato adicional necesitamos los patrones de corte que nuestra maquinaria es capaz realizar para la creación de las láminas demandadas. Estos patrones están determinados por la industria que quiere solucionar el problema. A veces es más costoso hacer el corte que el material desperdiciado por él, por lo que se optará por efectuar los mínimos cortes posibles aunque por ello pierda más material. En otras ocasiones ocurre todo lo contrario, el material es mucho más caro que la realización del corte, en estos casos generalmente, además de resolver el problema de corte, se debe resolver el problema de generación de todos los patrones posibles de manera de desperdiciar lo mínimo posible de material aunque se deban hacer muchos portes por cada lámina.

El problema genérico busca minimizar la cantidad de láminas del "stock" utilizadas y satisfacer la demanda de piezas.

El modelo es similar al caso de una dimensión:

$$\begin{aligned} \text{Min } & \vec{x} \\ \text{s.a. } & A\vec{x} \geq \vec{n} \\ & \vec{x} \geq 0 \end{aligned}$$

Donde:

- $\vec{x}$ representa un vector columna de las variables x_j, que indican la cantidad de veces que se utiliza el patrón j.
- A representa la matriz de m filas, donde cada columna $[a_1, a_2, ..., a_m]^T$, representa un patrón de corte rectangular. El elemento a_1, por ejemplo, representa las láminas de área $l_1 \times w_i$ que se generan.
- $\vec{n}$ representa el vector columna de las demandas N_i.

El problema de corte tiene variantes dependiendo del contexto como mencionábamos anteriormente, para nuestro caso nos enfocaremos en el "The assortment problem" (Problema de selección), es cual busca minimizar el número de láminas del stock a utilizar.

Básicamente el planteamiento a este problema es el modelo que hemos presentado, como extensión mostraremos los casos en que la maquinaria sólo puede realizar ciertos tipos de cortes por lo que tomaremos el tema de la incidencia de los patrones de corte en la solución óptima.

III-C. Extensión

El modelo de extensión es el siguiente:

$$\begin{aligned} \text{Min } & \vec{c}\vec{x} \\ \text{s.a. } & A\vec{x} \geq \vec{n} \\ & \vec{x} \leq \vec{v} \\ & \vec{x} \geq 0 \end{aligned}$$

Donde:

- $\vec{x}$ representa un vector columna de las variables x_j, que indican la cantidad de veces que se utiliza el patrón j.
- $\vec{c}$ representa un vector columna con los costos c_j asociados a cada uno de los patrones de corte x_j.
- A representa la matriz de m filas, donde cada columna $[a_1, a_2, ..., a_m]^T$, representa un patrón de corte rectangular. El elemento a_1, por ejemplo, representa las láminas de área $l_1 \times w_i$ que se generan.
- $\vec{n}$ representa el vector columna de las demandas N_i.
- $\vec{v}$: representa un vector con los límites de veces v_j en las que un patrón x_j puede ser ejecutado.

Básicamente, la extensión consiste en un nuevo parámetro a considerar y una nueva restricción, es decir, nuestro problema ahora considera dos factores importantes en la industria: tiempo y costos.

Los costos c_j representan que tanto le cuesta a la maquinaria realizar el patrón a_j, esto puede ser considerado en términos de energía, mano de obra necesaria, etc. Independiente de ello, este factor cambia las preferencias que antes teníamos, y la función objetivo ahora quiere minimizar los costos asociados a la generación de elementos más pequeños, en vez de reducir gastos por la utilización de materia prima.

La restricción $\vec{x} \leq \vec{v}$ introduce otro factor asociado al tiempo que puede demorar la maquinaria en realizar un corte, por lo tanto, hay que tener en cuenta la cantidad de veces máxima que podemos "ejecutar" el corte de un patrón, sobre todo cuando hay un tiempo límite en la que un pedido de producción debe ser satisfecho.

Como vemos, el modelo es más restrictivo, pero se acerca más a una situación real con respecto al planteamiento estándar, normalmente querremos acercarnos lo más posible a la situación por lo que lo normal es que se agreguen este tipo de extensiones.

IV. Experimentación

IV-A. *Entorno (Hardware y Software)*

Detalles entorno de experimentación:

- Software utilizado: LINGO v 10.0, 2006 [1]
- Hadware:
  - CPU: Intel Pentium G620 @ 2.60GHz
  - RAM: 4,00 GB DD3 @ 532 MHz
- Configuración del software: Por defecto.

Problema experimental (Parámetros del modelo)

Para la experimentación propondremos un problema simplificado para el modelo estándar y agregaremos detalles para el modelo extendido:

Armando muebles para computadoras

Supongamos que una empresa desea armar muebles para computadores, las cuales se componen principalmente de tablas de madera de distintas dimensiones. La empresa desea optimizar la generación de estas piezas utilizando la menor cantidad de materia prima posible. La información de la materia prima y de los componentes a generar se detallan a continuación:

1. Dimensión tabla "Grande": $100[cm] \times 100[cm]$
2. Demanda: 100 muebles
3. Tablas de madera para construir 1 mueble:
   - $a1 = 45[cm] \times 100[cm]$ (x 1)
   - $a2 = 45[cm] \times 75[cm]$ (x 3)
   - $a3 = 25[cm] \times 100[cm]$ (x 1)
   - $a4 = 45[cm] \times 65[cm]$ (x 1)
   - $a5 = 45[cm] \times 35[cm]$ (x 1)
4. Patrones de corte:
   - $x1 = a4$ (x2), $a3$ (x1), $a5$ (x1)
   - $x2 = a4$ (x1), $a1$ (x1), $a3$ (x1)
   - $x3 = a4$ (x1), $a2$ (x2)
   - $x4 = a1$ (x1), $a3$ (x1), $a4$ (x1)
   - $x5 = a2$ (x1), $a3$ (x2), $a5$ (x1)
   - $x6 = a3$ (x2), $a5$ (x2)
   - $x7 = a5$ (x2), $a2$ (x2)
   - $x8 = a5$ (x4), $a2$ (x1)
   - $x9 = a5$ (x3), $a3$ (x2)
   - $x10 = a1$ (x1), $a2$ (x1), $a5$ (x1)

Y para la extensión agregamos:

1. Costos de ejecutar cada patrón:
   - $x1 = 7$
   - $x2 = 4$
   - $x3 = 10$
   - $x4 = 2$
   - $x5 = 6$
   - $x6 = 7$
   - $x7 = 5$
   - $x8 = 3$
   - $x9 = 6$
   - $x10 = 9$
2. Límite de ejecución de un patrón (diarios):
   - $x1 = 70$
   - $x2 = 25$
   - $x3 = 50$
   - $x4 = 83$
   - $x5 = 71$
   - $x6 = 37$
   - $x7 = 65$
   - $x8 = 59$
   - $x9 = 44$
   - $x10 = 40$

IV-B. *Resultados modelo estándar*

A continuación mostramos el resultado obtenido por el software para el problema propuesto:

```
Global optimal solution found.
Objective value:  200.0000
Total solver iterations: 2


Variable  Value       Reduced Cost
X1        0.000000    1.000000
X2        0.000000    0.5000000
X3        100.0000    0.000000
X4        0.000000    0.5000000
X5        0.000000    0.5000000
X6        0.000000    1.000000
X7        0.000000    0.000000
X8        0.000000    0.5000000
X9        0.000000    1.000000
X10        100.0000   0.000000
```

La resolución del tiempo en el software es en segundos, por lo tanto no pudimos obtener el tiempo exacto de ejecución.

IV-C. *Resultados modelo extendido*

Para el modelo extendido el software nos da el siguiente resultado:

```
Global optimal solution found.
Objective value: 1291.000
Total solver iterations: 3
```

[1]Proporcionado para el curso vía moodle

Variable	Value	Reduced Cost
X1	0.000000	7.000000
X2	6.000000	0.000000
X3	50.00000	0.000000
X4	83.00000	0.000000
X5	0.000000	1.000000
X6	0.000000	7.000000
X7	65.00000	0.000000
X8	59.00000	0.000000
X9	0.000000	6.000000
x10	11.00000	0.000000

Como mencionamos anteriormente, la resolución de tiempo del software no alcanza para registrar el tiempo de ejecución.

V. Análisis de Resultados

En el modelo estándar vemos como el software escoge el mínimo número de patrones que generan las tablas solicitadas y de estas suplir la demanda, aparentemente muestra una preferencia por los patrones que abarcan o cubren más sub-elementos.

En el segundo caso, dada las restricciones agregadas, las preferencias cambian lógicamente acorde al beneficio que se espera y a las limitantes que acompañan a la ejecución de cada patrón.

Si bien no son comparables los resultados de las funciones objetivo dado que apuntan a cosas distintas, estas no son de naturaleza tan lejana, dado que implícitamente apuntan a abaratar costos de producción.

El cambio de un modelo a otro pasa simplemente por lo que se considere más relevante en la cadena de producción de sub-elementos, por lo que de ello dependerá las preferencias que al final nos muestre el resultado de la resolución del problema.

VI. Conclusiones y trabajo futuro

Parte del trabajo futuro consiste en la generación de patrones. Es aquí donde radica la raíz del problema de corte o más bien gran parte de su complejidad. Sin embargo, de omitir este paso y considerar que los patrones de corte ya están establecidos, el problema de optimización tomando en cuenta ciertas características (tipo de corte, coste asociado, uso de materia prima, etc.) está prácticamente resuelto, quedando en función del contexto las restricciones y los parámetros involucrados.

En general, en la literatura del problema, y en particular algunos de los autores mencionados en la sección II, hacen referencia a algoritmos para la generación de patrones y la optimización de los mismos, sin embargo sólo hicimos mención de ellos debido a que tienen una gran complejidad y su estudio es más interesante como un problema de combinatoria que como un problema de investigación de operaciones. Aún así la escala de la mano-factura actual hace que la cantidad de piezas solicitadas superen los cientos, en este contexto, los algoritmos que generan buenos patrones en tiempos acotados son de vital importancia.

Por otro lado, se puede solucionar el problema de corte con modelos que no sean de programación lineal. La programación dinámica, las heurísticas y las meta-heurísticas son ampliamente estudiadas para los problemas de optimización y tienen gran valor como instrumento de estudio para este tipo de problemas.

VII. Referencias

[1] Tzu-Yi Yu, Jiann-Cherng Yang, Yeong-Lin Lai, Chih-Cheng Chen and Han-Yu Chang. *Applying an Enhanced Heuristic Algorithm to a Constrained Two-Dimensional Cutting Stock Problem.* 26 Mar. 2014. http://www.naturalspublishing.com/files/published/7z5uf8oq7y668r.pdf

[2] Rosa Sumactika Delgadillo Avila. *Un estudio algorítmico del problema de corte y empaquetado 2D.* Lima, Perú 2007. http://cybertesis.unmsm.edu.pe/bitstream/cybertesis/1518/1/delgadillo_ar.pdf

[3] Richard M. Karp. *Reducibility Among Combinatorial Problems*

[4] P. Gilmore and R. Gomory. *A linear programming approach to the cutting stock problem*.

[5] P. Gilmore and R. Gomory. *A linear programming approach to the cutting stock problem - Part II*.

[6] P. Gilmore and R. Gomory. *Multistage Cutting Stock Problems of Two and More Dimensions*.

[7] J. E. Beasley. *Algorithms for unconstrained two-dimensional guillotine cutting*. Journal of the Operational Research Society, 1985.

[8] N Christofides, C Whitlock. *An algorithm for two-dimensional cutting problems*. Operations Research, 1977.

[9] KV Viswanathan, A Bagchi. *Best-first search methods for constrained two-dimensional cutting stock problems*. Operations Research, 1993.

[10] M Hifi, R Ouafi. *Best-first search and dynamic programming methods for cutting problems: the cases of one or more stock plates*. Computers & industrial engineering, 1997.

[11] PY Wang *Two algorithms for constrained two-dimensional cutting stock problems*. Operations Research, 1983.

[12] FJ Vasko. *A computational improvement to Wang's two-dimensional cutting stock algorithm*. Computers & industrial engineering, 1989.

[13] J. F. Oliveira, J. S. Ferreira. *An improved version of Wang's algorithm for two-dimensional cutting problems*. European Journal of Operational Research, 1990.

[14] A. P. Guevara, J. M. T. Goerlich, R. A Valdés. *Algoritmos Grasp y Tabu Search para el problema de corte guillotina bidimendional*. XXV Congreso Nacional de Estadística e Investigación Operativa: Vigo, 2000.

[15] F. Golver. *Tabu search and adaptive memory programming — advances, applications and challenges*. Interfaces in computer science and operations research, 1997.

[16] T. A. Feo, M. G. C. Resende. *A probabilistic heuristic for computationally difficult set covering problems*. Operation Research Letters, 1989.

[17] J. E. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. 1975.

[18] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. *Optimization by simmulated annealing*. science, 1983.

[19] V. Černý. *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*. Journal of optimization theory and applications, 1985.

[20] K. K. Lai, J. W. M. Chan. *Developing a simulated annealing algorithm for the cutting stock problem*. Computers & industrial engineering, 1997.

[21] L. Faina. *An application of simulated annealing to the cutting stock problem*. European Journal of Operational Research, 1999.

[22] V. Parada, M. Sepúlveda, M. Solar, A. Gómes. *Solution for the constrained guillotine cutting problem by simulated annealing*. Computers & Operations Research, 1998.

[23] S. Jakobs. *On genetic algorithms for the packing of polygons*. European Journal of Operational Research, 1996.

[24] Wu Tai-Hsi, Wu Yi-Hua, Chang Chin-Ju. *Solving Two-Dimensional Packing Problems by Using a Genetic Algorithm.*. Journal of Science and Engineering Technology, 2006.

[25] JF Gonçalves. *A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem*. European Journal of Operational Research, 2007.

[26] H. Dyckhoff. *A typology of cutting and packing problems*. European Journal of Operational Research, 1990.