

Resolución de problemas de corte con guillotina con métodos exactos y con algoritmos metaheurísticos

Alberto Gómez¹, David de la Fuente¹, Joaquín Bautista², Jorge Pereira²

¹ Dpto. de Admón. de Empresas y Contabilidad. Escuela Politécnica Superior de Ingeniería de Gijón. Campus de Viesques, 33204 Gijón (Asturias). agomez@epsig.uniovi.es, david@uniovi.es (Times NR, 10 puntos)

² Universidad Politécnica de Cataluña, Barcelona (Cataluña). joaquin.bautista@upc.edu

Resumen

El objetivo de este trabajo es presentar un algoritmo exacto y dos algoritmos heurísticos diseñados para optimizar el corte de chapa en una máquina de guillotina en L. Se persigue conseguir minimizar el número de chapas empleadas en la fabricación de una serie de piezas rectangulares. En el trabajo se comparan con la ayuda de un banco de ejemplos los resultados proporcionados por los algoritmos diseñados. Los algoritmos ofrecen muy buenos resultados aunque cada uno de ellos tiene sus peculiaridades que se presentan en el trabajo completo..

Palabras clave: programación entera, algoritmos genéticos, recocido simulado.

1. Introducción

Este trabajo se busca realizar un programa informático que permita minimizar el número de chapas de acero utilizadas por una empresa en su proceso productivo. En concreto, la aplicación diseñada se centra en optimizar la utilización de la guillotina tipo L, que la empresa objeto del estudio posee. Esta guillotina permite la realización de cortes horizontales y verticales simultáneamente, la razón de esta posibilidad reside en la forma de corte en L que tiene. Esta característica posibilita una utilización más eficiente de las chapas.

La forma de operar de la empresa es la siguiente: se comienza recibiendo de los clientes unas peticiones para el suministro de piezas, y a partir de dichas peticiones, se elaboran la orden de fabricación y el control numérico de la guillotina. Para la obtención de las piezas se emplean una superficie base de forma rectangular (en lo sucesivo chapa), de dimensiones fijas (2995mm*1250mm).

Basándose en estas premisas, el programa a implementar debe ubicar dentro de la chapa, las diferentes piezas solicitadas por los clientes, sabiendo que de cada tipo de pieza se suele pedir más de una unidad y que las piezas son todas de formas rectangulares. Además, normalmente se va a necesitar más de una chapa para poder satisfacer un determinado pedido. Al realizar la ubicación se busca minimizar el desperdicio de material, o lo que es lo mismo minimizar el número de chapas utilizadas. En la literatura, este tipo de problemas recibe el nombre de problema de *packing* multichapa.

2. Aproximación al problema de *packing*

La razón de este problema reside en que la materia prima (chapa) empleada en la industria metálica suele estar disponible en ciertos tamaños estándares; y normalmente es necesario cortar estas chapas en piezas de dimensiones menores antes de poder utilizarlas en el resto del proceso productivo. Como consecuencia de esta situación, la industria ha buscado mecanismos que le permitan maximizar la materia prima. Entre las primeras investigaciones efectuadas en este terreno, cabe destacar las realizadas por Gilmore y Gomory (1961,1963) para la resolución de problemas en una dimensión, y la ampliación que estos mismos autores realizaron en 1965 para problemas en dos dimensiones (Gilmore y Gomory, 1965). Desde entonces, el análisis de estos problemas se ha extendido rápidamente; sin embargo, no existe una solución global para todos ellos, debido a la gran complejidad que presentan.

Una forma de encontrar la solución a esta tipología de casos, es dividir el problema en varios subproblemas (Dyckhoff, 1990), e intentar resolver cada uno de ellos por separado. Precisamente, este trabajo se concentra en uno de estos subproblemas, el *packing*. Su resolución ha sido abordada por diversos autores, no llegando ninguno de ellos a encontrar un método que proporcione la solución óptima en todos los casos (Coffman et al, 1990; Jakobs, 1996; Steinberg, 1997; Csirik et al, 1997).

3. Descripción del problema

Como paso previo al análisis de la solución que se propone, conviene comentar con mayor amplitud las características del problema que se pretende resolver.

En primer lugar, destacar que no se permiten chapas de dimensiones diferentes a las estándar, todas las chapas tienen medidas estándar. En segundo lugar, y debido a las grandes dimensiones que presentan las piezas solicitadas por los clientes, en cada chapa el número de piezas a ubicar por lo general es reducido y suele oscilar entre 5 y 20 piezas.

Además, debido a que cada orden de fabricación suele estar formada por varios cientos de piezas, y teniendo en cuenta las grandes dimensiones de estas piezas, suelen ser necesarias más de 50 chapas para poder producir todas las piezas.

Por último, mencionar que dadas las condiciones particulares de la guillotina en L utilizada en esta empresa, se producen sobre la superficie base una serie de pérdidas, cuya importancia conviene destacar. En su primera aproximación a la chapa, la guillotina realiza dos cortes de aproximación, uno vertical y otro horizontal ambos de 5 mm. A su vez, el sistema de pinzas utilizado para agarrar la superficie base genera que se pierdan otros 20 mm horizontales. En la figura 1 se representa las pérdidas de material como consecuencia de la utilización de la guillotina en "L" objeto del presente trabajo. Las superficies base pasan a tener unas dimensiones de: $[(2995-5)\text{mm}*(1250-20-5)\text{mm}]$.

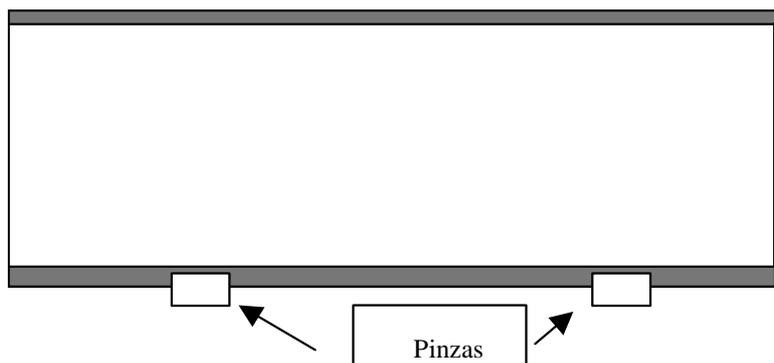


Figura 1: Material desperdiciado

Una vez comentadas las particularidades del problema analizado, se procede a describir a continuación los dos algoritmos de ubicación diseñados para su resolución.

4. Solución propuesta para el corte

4.1. Modelo exacto

A continuación, se comenta el modelo exacto desarrollado para resolver este apartado. Dado un número ilimitado de placas rectangulares idénticas que llamaremos bins de anchura W y altura H , el objetivo es situar (cortar) todos los ítems (piezas rectangulares) minimizando el número de bins.

Nomenclatura:

Parámetros

J	conjunto de ítems rectangulares $J=\{1,\dots,n\}$
w_j	Anchura del ítem $j \in J$
h_j	Altura del ítem $j \in J$
UB	Cota superior del número de bins requerido para cortar todos los ítems (se puede obtener con un procedimiento heurístico)
W	Anchura de un bin
H	Altura de un bin
a_{jpqrs}	=1 si el ítem j cuando se sitúa en posición normal con su vértice inferior izquierdo en el punto (p,q) cubre el punto (r,s) del bin a cortar. =0 en caso contrario
b_{jpqrs}	=1 si el ítem j cuando se sitúa en posición girada 90° con su vértice inferior izquierdo en el punto (p,q) cubre el punto (r,s) del bin a cortar. =0 en caso contrario

Variables:

x_{jkpq}	=1 si el ítem j se sitúa en posición normal con su vértice inferior izquierdo en el punto (p,q) del bin k ($1 \leq k \leq UB$) =0 en caso contrario
y_{jkpq}	=1 si el ítem j se sitúa en posición girada 90° con su vértice inferior izquierdo en el punto (p,q) del bin k ($1 \leq k \leq UB$) =0 en caso contrario
z_k	=1 si se utiliza el bin k ($1 \leq k \leq UB$) =0 en caso contrario

En tales condiciones, podemos escribir el siguiente ILP:

$$\min m = \sum_{k=1}^{UB} z_k \quad (0)$$

$$\sum_{k=1}^{UB} \left(\sum_{p=0}^{W-w_j} \sum_{q=0}^{H-h_j} x_{jkpq} + \sum_{p=0}^{W-h_j} \sum_{q=0}^{H-w_j} y_{jkpq} \right) = 1 \quad 1 \leq j \leq n \quad (1)$$

$$\sum_{j=1}^n \left(\sum_{p=0}^{W-w_j} \sum_{q=0}^{H-h_j} x_{jkpq} + \sum_{p=0}^{W-h_j} \sum_{q=0}^{H-w_j} y_{jkpq} \right) \leq n z_k \quad 1 \leq k \leq UB \quad (2)$$

$$\sum_{j=1}^n \left(\sum_{p=0}^{W-w_j} \sum_{q=0}^{H-h_j} a_{jpqrs} x_{jkpq} + \sum_{p=0}^{W-h_j} \sum_{q=0}^{H-w_j} b_{jpqrs} y_{jkpq} \right) \leq 1 \quad \begin{array}{l} 0 \leq r \leq W-1 \\ 0 \leq s \leq H-1 \end{array} \quad (3)$$

$$x_{jkpq} \in \{0,1\} \quad (1 \leq j \leq n) \wedge (1 \leq k \leq UB) \wedge (0 \leq p \leq W-w_j) \wedge (0 \leq q \leq H-h_j) \quad (4)$$

$$y_{jkpq} \in \{0,1\} \quad (1 \leq j \leq n) \wedge (1 \leq k \leq UB) \wedge (0 \leq p \leq W-h_j) \wedge (0 \leq q \leq H-w_j) \quad (5)$$

$$z_k \in \{0,1\} \quad 1 \leq k \leq UB \quad (6)$$

En el modelo, la función (0) expresa que el objetivo es minimizar el número de bins a utilizar. Las igualdades (1) obligan a situar en un único bin y en una única posición y orientación los items j ($j=1..n$). Las inecuaciones (2) se emplean para conocer si un bin se utiliza o no. Las inecuaciones (3) expresan que cualquier punto (r,s) de un bin puede estar cubierto por un ítem como máximo. Finalmente, las expresiones (4), (5) y (6) sirven para definir como binarias las variables del modelo.

El modelo presentado está basado en la propuesta de Beasley (1985). Se han añadido las variables asociadas a la asignación o no de cada pieza, girada 90° respecto a su posición normal, en cada bin disponible. Los elementos principales de dicho modelo, junto con otros posibles enfoques de modelización, pueden encontrarse también en Lodi, Martello and Monaci (2002).

4.2. Modelo heurístico

En este apartado se van a analizar los dos algoritmos diseñados para minimizar el número de chapas empleadas en el proceso productivo. El primero se basa en algoritmos genéticos y el segundo en recocido simulado. Sus principios de funcionamiento son semejantes: la ubicación inicial de las piezas en las chapas la realiza una heurística denominada *caída libre con reemplazo* y del afinado de la solución o soluciones iniciales se encarga un Algoritmo Genético (Goldberg, 1989) o un Recocido Simulado (Kirkpatrick et al, 1983). A continuación se explica con mayor detalle estas heurísticas.

En ambos algoritmos, la codificación empleada se basa en números enteros, se asigna a cada pieza, a colocar, un número entero, y el individuo solución se forma con una cadena de números (cadena de piezas). El orden en que aparecen los números, en la cadena, representa las posiciones en la chapa de las piezas.

Con objeto de explicar mejor la codificación, se comenta un ejemplo: considérese que se tienen 4 piezas rectangulares, para introducir en una chapa, a cada una de estas piezas se le asigna un número correlativo, a la primera pieza se le asigna el 1, a la segunda el 2 y así sucesivamente. El individuo solución (1,3,4,2) podría ser una solución al problema, esta solución implica que la pieza 1 es la primera en introducirse en la superficie base, seguida de la 3, de la 4, y finalmente de la 2.

En el caso analizado en este trabajo, hace falta modificar esta interpretación del individuo solución pues, como ya se ha comentado, no todas las piezas encajan en la misma chapa, y son necesarias varias chapas, para poder generar todas las piezas solicitadas por los clientes.

La asignación de una pieza a una chapa se realiza a la hora de analizar el individuo solución. Al leer la primera posición del individuo (primera pieza), se “activa” una chapa, y se ubica la pieza en la chapa; a continuación se lee la segunda posición del individuo (segunda pieza) y, se analiza si esta pieza se puede ubicar en la chapa o no, en caso afirmativo se asigna la pieza a la chapa y se lee la tercera posición; este proceso de lectura continua hasta que una de las piezas no se puede ubicar en la chapa. Cuando se detecta que no es posible asignar una pieza a la chapa, por no quedar un hueco suficientemente grande en ella, se “activa” una segunda chapa y se asigna la pieza a esta segunda chapa, este proceso prosigue así hasta que se hayan colocado todas las piezas. Con este sistema se determina el número de chapas necesarias para producir todas las piezas.

5. Resultados

Inicialmente, se planteó realizar pruebas con el modelo exacto propuesto en el apartado 4, pero se descubrió que con una versión simplificada del ejemplo más sencillo empleado en la empresa (el ejemplo de 16 piezas) y se tenían 1.937.664 variables. Ante esta dificultad, se intentó reducir el tamaño del problema con técnicas encaminadas a buscar máximo común divisor de los parámetros, sin embargo los resultados alcanzados fueron reducidos pues las dimensiones de los rectángulos no son fácilmente divisibles.

Ante estas dificultades, para el análisis de las soluciones propuestas se ha dispuesto del programa de optimización de la chapa utilizado anteriormente por la empresa para el control de la guillotina. Se ha generado un banco de quince ejemplos, y se ha comparado las soluciones alcanzadas por los diferentes algoritmos. En la tabla 1 se ofrece una comparativa de los resultados de los tres sistemas.

Los parámetros utilizados durante los diferentes ensayos por el algoritmo genético son: probabilidad de cruce, igual a 0,7; probabilidad de mutación: 0,3; probabilidad de mutación especial: 0,4; número de generaciones: 3 y tamaño de la población: 10 (con estos parámetros sólo se van a realizar 30 cálculos). Igualmente, los parámetros del recocido simulado se seleccionaron de forma que el número de cálculos que se realiza sea 30. El reducido valor del número de generaciones y del tamaño de la población viene impuesto por el funcionamiento solicitado. La empresa quiere un sistema “rápido”, establece que en unos tres minutos los algoritmos deben generar una solución equiparable a la proporcionada por el sistema original para el ejemplo 1 de la tabla 1.

	Nº	Algoritmo	Aleatorio RS		
Ejemplo	piezas	antiguo		AG	
1	547	69	71	61	67
2	229	45	48	47	45
3	52	21	21	21	21
4	16	5	4	5	4
5	228	42	45	42	42
6	722	110	107	91	103
7	683	151	149	144	149

8	732	176	175	174	173
9	798	114	115	105	111
10	754	136	133	129	131
11	580	82	79	79	79
12	1346	223	219	208	208
13	192	42	42	44	42
14	573	53	57	51	52
15	529	92	89	91	87

Tabla 1: Resultados de los experimentos. En la columna dos, se indica el número de piezas que forman el experimento y en las otras cuatro columnas se puede ver el número de chapas que se necesitan utilizar para poder obtener dichas piezas de acuerdo con cada uno de los algoritmos analizados.

En el Tabla 1, se puede apreciar como las heurísticas que se propone, ofrecen en la mayoría de las situaciones resultados mejores o cuando menos iguales a los ofrecidos por el programa original de la guillotina (algoritmo antiguo). Se observa que cuanto mayor es el número de piezas a ubicar mejores son las soluciones alcanzadas, tanto por el algoritmo genético como por el recocido simulado.

Con vistas a comprobar si realmente resulta necesaria la incorporación de los AGs y del RS. Se han realizado pruebas de generación de soluciones aleatorias (columna 3 de la tabla 1). La solución **aleatoria** se obtiene generando 30 soluciones aleatorias y analizando, con ayuda de la heurística de *caída libre con reemplazo*, el número de chapas necesarias para ubicar todos los rectángulos. De los resultados de esta columna, se aprecia la necesidad de los AGs o del RS.

Asimismo, en la tabla 1 se comprueba como no es posible discernir cuál de las dos heurísticas propuestas es mejor. Pues, en unos casos, el AG proporciona mejores soluciones y en otros ejemplos resulta más apropiado emplear RS.

6. Conclusiones

En el trabajo se presentan dos sistemas híbridos para la resolución de un problema industrial real, consistente en la automatización del funcionamiento de una guillotina tipo L. Esta automatización implica el diseño de un programa, el cuál distribuya dentro de una serie de chapas rectangulares, las piezas solicitadas por los clientes, de forma tal que se minimice el número de chapas utilizadas. Además, dicho programa debe indicar al control numérico de la guillotina, la secuencia en la cuál se debe cortar las diferentes piezas.

Para comprobar la calidad de las soluciones proporcionadas por los algoritmos, se han planteado una serie de ejemplos, y se ha comparado la calidad de las soluciones que ofrecen los algoritmos diseñados con las soluciones que proporciona el programa suministrado por el fabricante de la máquina, comprobándose que los algoritmos propuestos tiene un funcionamiento mejor en la mayoría de los casos. Asimismo, se han comparado entre si los dos algoritmos propuestos, de cara a determinar cuál de ellos era más idóneo para estos problemas, no pudiendo llegarse a ninguna conclusión, pues su idoneidad depende del ejemplo utilizado.

Referencias

- Coffman, E. G., Shor, P.W. (1990): Average-case analysis of cutting and packing in two dimensions. *Eur. J. O. R.* 44, 134-144.
- Csirik, J., Woeginger, G. J. (1997): Shelf algorithms for on-line strip packing. *Information Processing Letters.* 63, 171-175.
- Davis, L. (1991): Handbook of Genetic Algorithms. Van Nostrand Reinhold.
- Dyckhoff, H. (1990): A typology of cutting and packing problems. *European Journal of Operational Research.* 44, 145-159.
- Gilmore, P. C., Gomory, R. E. (1961): A linear programming approach to the cutting-stock problem. *Operations Research.* 9, 724-746.
- Gilmore, P. C., Gomory, R. E. (1963): A linear programming approach to the cutting stock problem. *Operations Research.* 11, 863-888.
- Gilmore, P. C., Gomory, R. E. (1965): Multi stage cutting stock problems of two and more dimensions. *Operations Research.* 13, 94-1120.
- Golberg, D. E., Lingle, R. (1985): Alleles, Loci, and the TSP. En *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ. 154-159.
- Goldberg, D. (1989): Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley Co., Inc, Reading, MA.
- Gómez, A., De la Fuente, D. (2000): Resolution of strip-packing problems with genetic algorithms. *Journal of the Operational Research Society.* 1289-1295.
- Gómez, A., De la Fuente, D., Priore, P. (2000): Resolución del problema de strip-packing mediante la metaheurística algoritmos genéticos. *Boletín de la SEIO.* 12-16.
- Gómez, A., Fernández, I., Parreño, J., García, N. (2001): Algoritmo genético para optimizar la utilización de material en guillotinas tipo L. XXVI Congreso Nacional de Estadística e Investigación Operativa. 102.
- Jakobs, S. (1996): On genetic algorithms for the packing of polygons. *Eur. J. O. R.* 88, 165-181.
- Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, M. P. (1983): Optimization by Simulated Annealing. *Science.* 220, 671-680.
- Steinberg, A. (1997): A Strip-Packing Algorithm with Absolute Performance Bound. *SIAM Journal on Computing.* 26, 2, 401-409.