Enfoques MILP para el tratamiento práctico del problema de patrones de corte en una dimensión

Gastón E. Salguero[1], María Analía Rodríguez[1], Juan M. Novas[2, 3]

[1] Instituto de Investigación y Desarrollo en Ingeniería de Procesos y Química Aplicada (IPQA)
Universidad Nacional de Córdoba-CONICET, Córdoba 5000, Argentina
[2] Centro de Investigación y Estudios de Matemática (CIEM)
Universidad Nacional de Córdoba-CONICET, Córdoba 5000, Argentina
[3] Centro de Investigación, Desarrollo y Transferencia de Sistemas de Información (CIDS)
Universidad Tecnológica Nacional- Facultad Córdoba, Córdoba 5000, Argentina
jmnovas@famaf.unc.edu.ar

Resumen. Con el fin de mejorar la competitividad, las industrias se encuentran en una permanente búsqueda de minimizar los desperdicios de sus procesos productivos. En el caso particular de los procesos de corte de materias primas para obtener bienes de menor tamaño, el cual se conoce como cutting stock problem (CSP), se intenta minimizar el desperdicio generado durante el procedimiento de corte. Por sus características combinatorias, este problema es de tipo NP-hard, por lo que diversas metodologías se han propuesto para su abordaje, entre las cuales se encuentran los modelos de programación matemática. En este trabajo, a partir de un modelo de optimización entero no lineal de la literatura, aplicado al CSP unidimensional, se propone un conjunto de enfoques novedosos basados en modelos matemáticos mixto entero lineales (MILP) que hacen foco en una correcta resolución de la problemática planteada y la eficiencia computacional. Todos los modelos presentados son analizados a partir de un mismo caso de estudio, que se utiliza a modo de ejemplificación, tomando en cuenta la dimensión de materia prima a utilizar, las medidas de los pedidos solicitados, y los límites máximos y mínimos de las cantidades de dichos pedidos.

Palabras clave: Problema de patrones de corte, minimización de desperdicio, generación de patrones, problema unidimensional.

1 Introducción

En las industrias se encuentra un objetivo común, transversal a todas las verticales: ser eficientes en el proceso productivo. Una de las estrategias para ello, es la búsqueda de reducción en los costos de producción, manteniendo los estándares requeridos. En la industria papelera, metalmecánica, del vidrio, textil, entre otras, se busca dar un máximo aprovechamiento a los recursos. Para el caso de la materia prima (MP), se persigue minimizar los desperdicios de la misma durante el proceso productivo. Existen diversas metodologías para abordar dicho objetivo, entre las cuales encontramos los modelos de programación matemática, provenientes del área de Investigación de Operaciones y hoy dentro del grupo de herramientas de Analítica Prescriptiva.

En el presente trabajo se indaga en el Problema de Patrones de Corte (CSP, por sus siglas en inglés). En el CSP se persigue minimizar el desperdicio (o scrap) y los demás costos asociados al proceso de corte, teniendo en cuenta las restricciones técnicas y de demanda que el sistema en cuestión impone [1], logrando así maximizar el uso de los recursos.

El CSP se caracteriza, según Dyckoff [2], por contar con dos elementos básicos: en primer lugar, un conjunto de piezas solicitadas por el cliente (ítems) que pueden ir variando su dimensión y, luego, un conjunto de materias primas (objetos) con dimensiones superiores. En el proceso de corte se asignan los ítems de menor dimensión a los objetos de mayores dimensiones, conformando patrones. Básicamente, como se menciona en [3], distintas cantidades conocidas de MP deben ser cortadas identificando los patrones de cortes factibles y buscando minimizar las partes residuales, que no pertenecen a los ítems. Estas son generalmente mencionadas como pérdida. Este trabajo trata el problema CSP desde distintos enfoques, y compara los resultados encontrados en cada caso. El trabajo se alinea con las conclusiones presentadas en Delorme y colab. [4], donde se plantea que las mejoras en el poder resolutivo de los solvers matemáticos estimula la búsqueda de nuevos algoritmos y métodos integrados. Se estudia de manera metodológica el problema CSP, unidimensional, a partir de un modelo de optimización lineal, explorando también alternativas con modelos MILP, con el fin de observar cómo estos acercamientos minimizan scrap y la cantidad de patrones diferentes.

2 Descripción del problema

Se analizó el desarrollo de Haessler y Sweeney [3], en donde inicialmente se plantea un modelo no lineal de minimización de scrap total en corte de rollos del mismo diámetro y diferente longitud, según diversos pedidos. Tal formulación no garantiza optimalidad. En dicho trabajo se presenta la diferenciación de objetos y de ítems, de modo tal que el objeto es la medida mayor de la cual serán cortados/extraídos conjuntos formados por ítems de distintos tamaños, los cuales conforman un patrón. Los ítems representan los productos solicitados por los clientes. Dicho modelo plantea límites máximos y mínimos en relación a la cantidad de productos solicitados. El problema es de tipo 1/V/I/R, según la clasificación presente en [2], la cual lo ubica como un problema CSP clásico.

Como el modelo inicial propuesto por Haessler y Sweeney [3] no garantizaba la optimalidad, los autores desarrollaron un enfoque sobre la problemática mediante programación lineal (LP, por sus siglas en inglés), con base en lo trabajado por Gilmore y Gomory [5]. Dicha contribución desglosa el problema en dos, incorporando una variable dual que se relaciona directamente con el número de ítems. Como indican los autores, la principal desventaja de utilizar LP de esta manera, es que el número de patrones de corte activos en la solución será muy cercano al número de ítems solicitados. Esto implica un aumento de la demanda de recursos y de la complejidad computacional.

3 Formulación de los modelos

Los modelos MILP propuestos en el presente trabajo, tienen como punto de partida al modelo desarrollado en [3]. Dicha formulación se describe en la sección 3.1.
La problemática ha sido trabajada desde tres abordajes distintos. En el primer caso, se presenta un modelo que utiliza un algoritmo heurístico para la generación de patrones. En el segundo enfoque, se enfrenta al problema resolviendo dos formulaciones de manera consecutiva. En el último modelo, se integran todas las variables a considerar. A continuación, luego de introducir la nomenclatura a emplear, se presentan el modelo original en [3] y cada una de las tres formulaciones propuestas.

Nomenclatura:

Conjuntos e índices:

I	Grupo de pedidos solicitado por un cliente	$i \in I$
J	Grupo de patrones	$j \in J$
K	Conjunto que indica la cantidad de veces que se repite un patrón	$k \in K$

Parámetros:

UW	Medida de materia prima, ancho útil
RLP	Límite inferior para que cada pedido i cuente con al menos un ítem (RLP=1) sumando entre todos los patrones
M	Big M
Cost	Representa el costo en el que se incurre a la hora de realizar un cambio de patrón
ScrapC	Representa el costo en el que se incurre en la generación de scrap de cada patrón
W_i	Longitud de los ítems requeridos para cada pedido i
RL_i	Cantidad mínima requerida del pedido i
RU_i	Cantidad máxima requerida del pedido i

Variables continuas:

Za	Cantidad total de scrap/desperdicio total.
Zb	Sumatoria de los desperdicios en los que incurre cada uno de los patrones
Zc	Costo total en el que se incurre entre el scrap y los cambios de patrones
$Tk_{j,k}$	Desperdicio/scrap en el que incurre el patrón j repetido k veces
T_j	Pérdida por recorte en la que incurre el patrón j

Variables enteras:

X_j	Número de objetos que son cortados usando patrón j
$A_{i,j}$	Cantidad de piezas del tipo i a cortar por el patrón j
$P_{i,j,k}$	Cantidad total de ítems i obtenidos producto de ejecutar k veces el patrón j

Variables binarias:

$Y_{i,j}$	Indica el uso o no de un elemento i en el patrón j
$S_{j,k}$	Indica la cantidad de ejecuciones k sobre el patrón j

$B1_{i,j,j'}$ Representa desigualdad entre las variables $A_{i,j}$ y $A_{i,j'}$; siendo igual a 1 cuando la primera es mayor

$B2_{i,j,j'}$ Representa desigualdad entre las variables $A_{i,j}$ y $A_{i,j'}$; siendo igual a 1 cuando la segunda es mayor

$B3_{i,j,j'}$ Representa valores nulos para las variables $A_{i,j}$ y $A_{i,j'}$; siendo igual a 1 cuando ambas son iguales a cero.

3.1 Modelo Original

Se toma como base el modelo original de Haessler y Sweeney [3], donde se formula el problema como se muestra a continuación.

$$\min Za = \sum_j T_j X_j \tag{1}$$

Sujeto a:

$$\sum_j A_{i,j} X_j \geq RL_i \; \forall i \tag{2}$$

$$\sum_j A_{i,j} X_j \leq RU_i \; \forall i \tag{3}$$

$$\sum_i A_{i,j} W_i \leq UW \; \forall j \tag{4}$$

$$T_j = UW - \sum_i A_{i,j} W_i \; \forall j \tag{5}$$

$$X_j \geq 0 \; \forall j \tag{6}$$

$$X_j \in \{enteros\} \tag{7}$$

$$A_{i,j} \geq 0 \; \forall i,j \tag{8}$$

$$A_{i,j} \in \{enteros\} \tag{9}$$

La variable Za de la función objetivo a minimizar (1), se define como la sumatoria entre todos los patrones empleados, del scrap de cada patrón *j*, por la cantidad de repeticiones del mismo. La Ec. (2) y la Ec. (3) aseguran, para cada pedido *i*, que la cantidad no sea menor, ni mayor, que el límite inferior RL_i y que el límite superior RU_i, respectivamente. El ancho de cada patrón *j*, conformado por el producto $A_{i,j} * W_i$, debe ser menor que el ancho del objeto, UW. La Ec. (4) asegura que ningún patrón exceda el ancho total de la materia prima. La Ec. (5) describe el desperdicio en el que incurre el patrón *j*. La restricción de positividad de X_j queda representada por la Ec. (6) y la restricción (7) establece que X_j pertenece a los enteros. La restricción de positividad de $A_{i,j}$ queda representada por la Ec. (8) y la Ec. (9) establece a $A_{i,j}$ como enteros.

3.2 Enfoque 1. Combinación de algoritmo más modelo

Este primer enfoque trabaja con un modelo que, en una etapa de preprocesamiento, para la generación de patrones, utiliza un algoritmo heurístico. El mismo, a partir de una lista de ítems i, presentada de manera decreciente en cuanto a su dimensión, puede ir generando patrones j, colocando dichos ítems en un patrón nuevo a medida que ya no pueden incorporarse en el patrón previo. De este modo, se van logrando patrones a partir de la cantidad de ítems, y repeticiones de los mismos, cuya suma de dimensiones no superan la dimensión del objeto de ancho UW. Luego, estos patrones son input de un modelo MILP que define la cantidad de repeticiones a utilizar de cada uno, considerando los límites máximos, RU_i, y límites mínimos, RL_i, indicados por el cliente para cada ítem, y a la vez que busca minimizar el scrap total generado.

El algoritmo que se presenta a continuación responde a la lógica de los algoritmos de mejor ajuste decreciente (BFD, por sus siglas en inglés) [4] y [6], según la cual cada ítem se acomoda en el objeto más lleno que lo pueda almacenar, incorporando nuevos objetos en caso de ser necesario, considerando un orden decreciente de las medidas i solicitadas.

Como input se ingresa una lista de distintos pedidos i ordenados de manera decreciente según su ancho. Para el desarrollo de este algoritmo se incorporan también los siguientes parámetros.

resto $_j$	Identifica el excedente útil a medida que se va conformando el patrón;
suma $_i$	Sumatoria de anchos de pedidos *i*. Se emplea para controlar que cada *i* no exceda su RU_i;
rp $_{i,j}$	Número de repeticiones del pedido *i* en el patrón *j*;
T_j^*	Desperdicio/scrap en el que incurre el patrón j.

Algoritmo

```
    suma_i = 0;
    rp_{i,j} = 0;
    Mientras i = j;
        Para cada i en I
          Para cada j en J
            A*_{i,j} = min (floor (UW / W_i), RU_i);
            resto_j = UW – A*_{i,j} * W_i;
            Para cada jj en J
                    suma_i = Σ_{jj} A*_{i,jj} ;
            Fin Para
            Para cada ii en I
               Mientras i < > ii & resto_j > 0 & W_ii < = resto_j & suma_ii < = RU_ii
                  Si rp_{i,j} < = 0 entonces
                       rp_{ii,jj} = rp_{ii,jj} +1;
                       A*_{ii,j} = min (floor (resto_j / W_ii), RU_ii);
```

resto $_{j}$ = resto $_{j}$ – $A^{*}_{ii,j}$ $*$ W_{ii};
Para cada jj en J
suma $_{ii}$ =$\sum_{jj} A^{*}_{ii,j}$
Fin Para
Fin Si
Fin Mientras
Fin Para
$T^{*}_{j} = UW - \sum_{ii} A_{ii,j} * W_{ii}$;
Fin Para
Fin Para
Fin Mientras
Fin Algoritmo

Una vez que se obtienen los patrones factibles a partir del algoritmo previo, se ingresa al modelo el parámetro $A^{*}_{i,j}$, que reemplaza a la variable $A_{i,j}$ del modelo original. Lo mismo sucede con T^{*}_{j}, que se ingresa como parámetro al modelo reemplazando a la variable T_{j}. Se debe tener en consideración que la función objetivo es igual a la Ec. (1) en la primera formulación con la diferencia que en este caso T^{*}_{j} ingresa como parámetro. Lo mismo sucede con $A^{*}_{i,j}$ en las Ecs. (2 – 4).

Se emplea la formulación del modelo original con las Ecs. (1), (2), (3), (4), (6) y (7) para resolver el problema, considerando los ajustes descriptos en párrafo previo. La función objetivo, entonces linealizada, queda como se muestro previamente.

3.3 Enfoque 2. Modelos secuenciales

El segundo enfoque trabaja con dos modelos que se ejecutan de manera consecutiva. El primero, se encarga de crear patrones buscando minimizar el scrap total generado por cada uno. El segundo modelo, determina qué patrones de los generados por el primer modelo serán utilizados minimizando el scrap total generado considerando patrones en uso y sus correspondientes repeticiones. En la primera formulación de este segundo enfoque se mantienen los índices, parámetros y variables utilizados en el Enfoque 1.

Para este enfoque, se desarrollaron dos alternativas.

Modelo de generación de patrones (versión 1)

Una primera versión del modelo, la cual mencionaremos como versión v1, incorpora además el parámetro escalar RLP y la variable $Y_{i,j}$. También, define una primera tanda (la mitad) de patrones en los cuales se cumple la condición de incorporar ítems solo si el orden de *i* es igual al orden de *j*. Luego, la segunda mitad de patrones son generados de acuerdo al mínimo desperdicio. El modelo está determinado por la función objetivo, dada por la Ec. (10), sujeto a las restricciones dadas por las Ecs. (4), (8) y (9), presentadas previamente, a las que se suman las Ecs. (11) a (15).

$$\min Zb = \sum_{j}\left(UW - \sum_{i} A_{i,j} W_i\right) \tag{10}$$

Sujeto a:

$$A_{i,j} \leq RU_i \; Y_{i,j} \; \forall \; i,j \tag{11}$$

$$Y_{i,j} = 1 \; \forall \; i,j \; \forall \; \mathrm{ord}(i) = \mathrm{ord}(j) \tag{12}$$

$$\sum_{j} A_{i,j} \geq 1 \; \forall \; i\,, \mathrm{ord}(j) > |I| \tag{13}$$

$$\sum_{i} Y_{i,j} = 1 \quad \forall \; j, ord(j) \leq |I| \tag{14}$$

$$Y_{i,j} \; \in \; \{0,1\} \tag{15}$$

La función objetivo (10) minimiza la sumatoria del scrap generado por cada patrón. La Ec. (11) representa la cantidad máxima de unidades *i* que pueden ser asignadas a un patrón *j*, lo cual está dado por la cota superior de su pedido RU_i . La Ec. (12) indica que el pedido *i* se asigna al patrón *j*, tal que la dupla *i, j* cuenten con el mismo número de orden, lo cual restringe que el orden de *i* coincida con orden de *j*. La Ec. (13) define límite inferior para que cada pedido *i* cuente con al menos un ítem en cada *j* dentro del rango indicado. La Ec. (14) obliga a que todos los patrones donde el orden de j es menor o igual al valor cardinal de I, cuenten con un solo tipo de ítem. La restricción (15) establece que $Y_{i,j}$ solo puede tomar los valores 0 o 1.

El Enfoque 2 v1 presenta una tendencia a la generación de múltiples patrones idénticos. Con el fin de evitar dichas repeticiones, se presenta la siguiente variación al mismo.

Modelo de generación de patrones (versión 2)

Para el Enfoque 2 v2, en el primer paso, el de generación de patrones, se incorporó el parámetro escalar big-M (18 a 21). Al igual que la versión 1, el modelo está determinado por la función objetivo, dada por la Ec. (10), sujeto a las restricciones dadas por las Ecs. (4), (9), (11), (14) y (15), presentadas previamente, a las que se suman las Ecs. (16) a (24). En este caso, no se consideran las restricciones (8), (12) y (13) y se agregaron las variables binarias $B1_{i,j,j'}$, $B2_{i,j,j'}$ y $B3_{i,j,j'}$. El primer modelo, de los dos secuenciales, queda sujeto a las siguientes restricciones adicionales:

$$Y_{i,j} \geq 1 \; \forall \; i,j \; \forall \; \mathrm{ord}(i) = \mathrm{ord}(j) \tag{16}$$

$$\sum_{i} A_{i,j} \geq 1 \; \forall \; j \tag{17}$$

$$A_{i,j} - A_{i,j'} \geq 1 - M\left(1 - B1_{i,j,j'}\right) \quad (18)$$
$$\forall\, i,j,j', \mathrm{ord}(j) \neq \mathrm{ord}(j'), \mathrm{ord}(j') > |I|$$

$$A_{i,j'} - A_{i,j} \geq 1 - M\left(1 - B2_{i,j,j'}\right) \quad (19)$$
$$\forall\, i,j,j', \mathrm{ord}(j) \neq \mathrm{ord}(j'), \mathrm{ord}(j') > |I|$$

$$A_{i,j} - A_{i,j'} \geq 0 - M\left(1 - B3_{i,j,j'}\right) \quad (20)$$
$$\forall\, i,j,j', \mathrm{ord}(j) \neq \mathrm{ord}(j'), \mathrm{ord}(j') > |I|$$

$$A_{i,j} - A_{i,j'} \leq 0 + M\left(1 - B3_{i,j,j'}\right) \quad (21)$$
$$\forall\, i,j,j', \mathrm{ord}(j) \neq \mathrm{ord}(j'), \mathrm{ord}(j') > |I|$$

$$\sum_{i}\left(B1_{i,j,j'} + B2_{i,j,j'}\right) \geq 1 \quad (22)$$
$$\forall\, j,j', \mathrm{ord}(j) \neq \mathrm{ord}(j)$$

$$B1_{i,j,j'} + B2_{i,j,j'} + B3_{i,j,j'} = 1 \quad (23)$$
$$\forall\, i,j,j', \mathrm{ord}(j) \neq \mathrm{ord}(j)$$

$$B1_{i,j,j'}, B2_{i,j,j'}, B3_{i,j,j'} \in \{0,1\} \quad (24)$$

En este caso la Ec. (16) indica que el pedido *i* se asigna al patrón *j*, tal que la dupla *i*, *j* cuenten con el mismo número de orden. La Ec. (17) asegura que en cada patrón haya uno o más ítems. La Ec. (18) determina que si la variable binaria $B1_{i,j,j'}$ es uno, entonces la cantidad de unidades de *i* asignadas al patrón *j* es mayor que la cantidad de unidades de *i* asignadas a *j'*. De manera similar, la Ec. (19) determina que, si la variable binaria $B2_{i,j,j'}$ es uno, entonces la cantidad de unidades de *i* asignadas a *j'*, es mayor que la cantidad de unidades de *i* asignadas a *j*. Las Ecs. (20) y (21) aseguran que, si $B3_{i,j,j'}$ es uno, entonces la cantidad de unidades del pedido *i* asignada al patrón *j* es igual a la cantidad asignada al patrón j', siendo incluso posible que sean nulas. La Ec. (22) determina que debe consignar algún ítem *i* a *j* o a *j'*. La Ec. (23) determina que se cumpla alguna de las tres condiciones y la Ec. (24) indica que $B1_{i,j,j'}$, $B2_{i,j,j'}$ y $B3_{i,j,j'}$ solo pueden tomar los valores 0 o 1.

Modelo para optimización de corte

Resuelta la etapa previa, a través de alguno de los modelos propuestos (v1 o v2), se toman los valores allí obtenidos para la variable $A_{i,j}$, y se los considera como input del segundo modelo, denominando a este parámetro $A^*_{i,j}$. También se calcula el parámetro T^*_j mediante el parámetro ingresado $A^*_{i,j}$ (adaptación de la Ec. 5 considerando $A^*_{i,j}$).

Considerando esto, el segundo modelo se implementa mediante las Ecs. (1) a (7), al igual que en el Enfoque 1.

3.4 Enfoque 3. Modelo integral

Este último enfoque propuesto, aborda de manera íntegra todas las variables a considerar. Además de ello, el acercamiento incorpora valores de costo por scrap generado y costo por el uso de un patrón distinto.

La incorporación de costos nos permite tener en cuenta dos aspectos a optimizar: (i) minimizar los costos ocasionados por el scrap total generado, y (ii) penalizar la generación de patrones. El modelo tiene como objetivo que dicha disminución de scrap se logre con el uso de la menor cantidad de patrones posibles. A fin de lograr un modelo que integre tanto la generación de patrones como la optimización del CSP sin incurrir en no linealidad, se introduce el concepto de repeticiones de patrones, el cual es manejado a través del elemento *k*, donde *k* indica la cantidad de veces que un dado patrón es repetido.

$$\min Zc = \sum_{j,k} Tk_{j,k}\ ScrapC + S_{j,k}\ Cost \tag{25}$$

Sujeto a:

$$RU_i\ Y_{j,k} \geq P_{i,j,k}\ \forall\, i,j,k \tag{26}$$

$$P_{i,j,k} \leq A_{i,j}\ k + RU_i\ k\ (1 - S_{j,k})\ \forall\, i,j,k \tag{27}$$

$$P_{i,j,k} \geq A_{i,j}\ k - RU_i\ k\ (1 - S_{j,k})\ \forall\, i,j,k \tag{28}$$

$$\sum_{k} S_{j,k} \leq 1\ \forall\, j \tag{29}$$

$$\sum_{j,k} P_{i,j,k} \geq RL_i\ \forall\, i \tag{30}$$

$$\sum_{j,k} P_{i,j,k} \leq RU_i\ \forall\, i \tag{31}$$

$$\sum_{k} S_{j,k}\ RU_i \geq A_{i,j}\ \forall\, i,j \tag{32}$$

$$Tk_{j,k} \geq T_j\ k - UW\ k\ (1 - S_{j,k})\ \forall\, j,k \tag{33}$$

$$Tk_{j,k} \leq T_j\ k + UW\ k\ (1 - S_{j,k})\ \forall\, j,k \tag{34}$$

$$Tk_{j,k} \geq 0\ \forall\, j,k \tag{35}$$

$$A_{i,j}, P_{i,j,k}, S_{j,k} \in \{enteros\} \tag{36}$$

La Ec. (25) define la función objetivo que minimiza el costo total en el que se incurre al producir de las piezas. La Ec. (26) garantiza que, si el patrón es utilizado, la cantidad total producidas del ítem *i* en el patrón *j* producto de *k* repeticiones no exceda el límite RU_i. Las Ecs. (27) y (28) determinan la cantidad de ítem *i* que se producen en el patrón *j* el cual se calcula en función de la cantidad de unidades del ítem asignadas al patrón multiplicado por la cantidad de veces *k* que el patrón *j* se repite. La Ec. (29) determina que solo se asigne como máximo un único número de repeticiones a cada patrón. Las Ecs. (30) y (31) restringen el límite inferior y superior (respectivamente) en los ítems *i*. La Ec. (32) indica que si el patrón *j* es utilizado, entonces, la cantidad de unidades del ítem *i* asignadas a *j* no debe exceder su límite superior. La Ec. (33) y (34) permiten calcular el scrap en el que incurre el patrón j repetido *k* veces, cuando el patrón es efectivamente utilizado. La Ec. (35) introduce una restricción de positividad de $Tk_{j,k}$. Por último, la Ec. (36) restringe la condición de integralidad sobre las variables $A_{i,j}, P_{i,j,k}, Y_{j,k}$.

Además, se incorpora como restricciones las Ec. (4) y (5), ya desarrolladas en el enfoque original.

4 Casos de Estudio y Resultados

Con el fin de analizar los resultados al emplear los modelos propuestos, se resuelven casos de estudio donde se necesita realizar cortes unidimensionales sobre perfiles de acero. Solo se considera la medida estándar de 600 cm. como longitud de dichos perfiles.

La cantidad de elementos considerados, en todas las instancias, han sido 20 ítems (los mismos para todos los ejemplos). Dichos ítems se generaron aleatoriamente con medidas de entre 200 y 590 cm. La amplitud del rango de valores para cada i, RL(i) a RU(i), se dispone que varíe en cuatro grupos: la diferencia entre la cantidad mínima y máxima solicitada por cada cliente, va de 40 productos, a 10 productos, según el grupo.

Las pruebas se realizaron con un tope máximo de 1800 segundos de CPU o en cuanto lograra llegar a gap 0.00, lo que sucediera primero. Se empleó el software GAMS versión 45.4.0 [7] y el solver Cplex 22.1.1.0. Se utilizó una computadora con procesador Intel Core i5-7400 CPU, CPU 3.0 GHz y 16GB RAM. En la Tabla 1 se presentan los resultados obtenidos al emplear los Enfoques 1 y 2, los cuales comparten la misma función objetivo.

Table 1. Resultados de Enfoques 1 y 2

RL -RU	Tipo de Modelo	Scrap Total	Patrones Generados/ en Uso	F.O. - Scrap Total	Ejecución en segundos
10 – 50	Enfoque 1	10530	20/16	10530	1.00
10 – 50	Enfoque 2 v1	10210	40/18	10210	0.57
10 – 50	Enfoque 2 v2	10210	40/18	10210	1801.57

20 – 50	Enfoque 1	21060	20/16	21060	0.17
20 – 50	Enfoque 2 v1	23580	40/19	23580	0.57
20 – 50	Enfoque 2 v2	20740	40/17	20740	1801.57
30 – 50	Enfoque 1	31590	20/16	31590	0.04
30 – 50	Enfoque 2 v1	43270	40/20	43270	0.50
30 – 50	Enfoque 2 v2	31270	40/18	31270	1801.54
40 – 50	Enfoque 1	42120	20/16	42120	0.07
40 – 50	Enfoque 2 v1	62960	40/20	62960	0.59
40 – 50	Enfoque 2 v2	41960	40/18	41960	1801.57

En todos los casos previamente desarrollados, tanto para el Enfoque 1 como para el Enfoque 2 v1, se obtuvo un gap relativo final de 0.00. Para el Enfoque 2 v2, se obtuvo un gap que rondó el 22% en la etapa de generación de patrones y un gap del 0% para la segunda etapa, de optimización del scrap.

Al indagar en el mejor rendimiento de tiempo, de los casos mostrados en Tabla 1, se ve logrado por Enfoque 1 y Enfoque 2 v1. Para el caso del Enfoque 1, se observa que en un instante resuelve un scrap aceptable, aunque no el mejor, en comparación al de los otros modelos, para los cuatro grupos. A su vez, utiliza la mínima cantidad de patrones. El mejor desempeño en relación al scrap lo logra el Enfoque 2 v2.

Se procede a analizar los resultados obtenidos considerando, y cotejando, los modelos previos (Enfoque 1 y Enfoques 2) con el Enfoque 3, cuya función objetivo incluye costos.

En esta oportunidad se utiliza el mismo caso de estudio previo. Se debe tener en consideración que, en todos los casos asentados en las Tabla 2 y Tabla 3, el RU es de 50 unidades para todos los ítems.

En la Tabla 2 se comparan los resultados obtenidos al emplear los Enfoques 1, 2 y 3, referidos al scrap total, los patrones generados y en uso, el GAP relativo del primer modelo (en caso de que corresponda) y del modelo final, y el tiempo de ejecución. La tabla permite comprender como, considerando que cuentan diferentes funciones objetivo, en el Enfoque 3 se logran valores de scrap prometedores a pesar de su GAP elevado.

Table 2. Desempeño de los distintos enfoques propuestos

RL	Tipo de Modelo	Scrap Total	Patrones Generados/ en Uso	GAP Relativo 1er./Final	Ejecución Total en seg.
10	Enfoque 1	10530	20/16	0.00/0.00	1.00
10	Enfoque 2 v1	10210	40/18	0.00/0.00	0.57
10	Enfoque 2 v2	10210	40/18	0.22/0.00	1801.57
10	Enfoque 3	10210	17/17	0.00/0.88	1800.34
20	Enfoque 1	21060	20/16	0.00/0.00	0.17

20	Enfoque 2 v1	23580	40/19	0.00/0.00	0.57
20	Enfoque 2 v2	20740	40/17	0.21/0.00	1801.57
20	Enfoque 3	20740	16/16	0.00/0.95	1800.34
30	Enfoque 1	31590	20/16	0.00/0.00	0.04
30	Enfoque 2 v1	43270	40/20	0.00/0.00	0.50
30	Enfoque 2 v2	31270	40/18	0.22/0.00	1801.54
30	Enfoque 3	31318	18/18	0.00/0.97	1803.12
40	Enfoque 1	42120	20/16	0.00/0.00	0.07
40	Enfoque 2 v1	62960	40/20	0.00/0.00	0.59
40	Enfoque 2 v2	41960	40/18	0.22/0.00	1801.57
40	Enfoque 3	42120	15/15	0.00/0.98	1800.70

Se logra apreciar que el mejor desempeño tanto en gap como en scrap generado se sigue dando por el Enfoque 2 v2, en el cual se ha condicionado la no repetición de patrones.

En el caso del Enfoque 3 se observa que, si bien se encuentra dentro de los márgenes de scrap mínimo, y utilizando la menor cantidad de patrones, el costo computacional es muy elevado y no se logra garantizar la calidad del resultado obtenido. La cantidad de patrones se ve optimizada en este modelo debido a que, a diferencia del resto, se busca minimizar el costo total scrap junto con el costo por cambio de patrón. El requerimiento computacional se torna elevado debido a que se contempla el uso de un tercer conjunto que nos permite calcular repeticiones de patrón; evitando de esta manera la no linealidad del modelo original, pero incurriendo en un tamaño de problema mayor, debido a la incorporación de dicho índice en las variables $Tk_{j,k}, P_{i,j,k}, Y_{j,k}$.

En la Tabla 3 se representa el tiempo que conlleva la generación de patrones, el tiempo que conlleva la asignación de patrones a la materia prima y el tiempo de ejecución total de cada uno de los tres enfoques. A su vez también se indica el GAP relativo, las cantidades de ecuaciones, variables continuas y variables discretas, involucradas en la ejecución del primer modelo (en caso de que corresponda) y del modelo final.

Se observa en la Tabla 3 que, en cuanto se emplea un modelo que analiza el espectro de posibilidades combinatorias, evitando así la repetición de patrones, se produce un aumento de ecuaciones y variables que ocasionan un aumento en el tiempo de ejecución. Tales son los casos del Enfoque 2 v2 y del Enfoque 3. En las instancias en las que no se garantiza diferencia de patrones o en el que los patrones son generados por algoritmos (Enfoque 1 y Enfoque 2 v1, respectivamente) se observa un muy buen desempeño en tiempos de ejecución, pero acompañado de un aumento del scrap.

Las distintas alternativas de solución presentadas y sus variaciones de performance respecto al tiempo de ejecución y resultados, permite contar con diferentes herramientas que el decisor puede emplear según sus requerimientos de tiempo, recursos y calidad de solución requerida.

Table 3. Comparativa de tamaño de los modelos.

RL	Tipo de Modelo	Generar Patrones en seg.	CSP en seg.	Total en seg.	GAP Relativo 1er./Final	Cantidad de Ec. 1er./Final	Variables Continuas 1er./Final	Variables Discretas 1er./Final
10	Enfoque 1	-	1.00	1.00	0.00/0.00	0/61	0/21	0/20
10	Enfoque 2 v1	0.42	0.15	0.57	0.00/0.00	2141/81	1/1	1640/40
10	Enfoque 2 v2	1800.01	1.56	1801.57	0.22/0.00	96081/81	1/1	95200/40
10	Enfoque 3	-	1800.34	1800.34	0.00/0.88	0/13722	0/222	0/4600
20	Enfoque 1	-	0.17	0.17	0.00/0.00	0/61	0/21	0/20
20	Enfoque 2 v1	0.42	0.15	0.57	0.00/0.00	2141/81	1/41	1640/40
20	Enfoque 2 v2	1800.01	1.56	1801.57	0.21/0.00	96081/81	1/41	95200/40
20	Enfoque 3	-	1800.34	1800.34	0.00/0.95	0/26522	0/422	0/8800
30	Enfoque 1	-	0.04	0.04	0.00/0.00	0/61	0/21	0/20
30	Enfoque 2 v1	0.42	0.15	0.50	0.00/0.00	2141/81	1/1	1640/40
30	Enfoque 2 v2	1800.01	1.56	1801.54	0.22/0.00	96081/81	1/1	95200/40
30	Enfoque 3	-	1803.12	1803.12	0.00/0.97	0/39322	0/622	0/13000
40	Enfoque 1	-	0.07	0.07	0.00/0.00	0/61	0/21	0/20
40	Enfoque 2 v1	0.42	0.15	0.59	0.00/0.00	2141/81	1/1	1640/40
40	Enfoque 2 v2	1800.01	1.56	1801.57	0.22/0.00	96081/81	1/1	95200/40
40	Enfoque 3	-	1800.70	1800.70	0.00/0.98	0/52122	0/822	0/17200

5 Conclusiones

Se ha abordado la problemática del CSP desde tres perspectivas que, si bien apuntan a la reducción del scrap, lo hacen desde distintos enfoques. Se comprende que la mayor variedad de patrones formulados favorece a que los modelos cuenten con más posibilidades de reducir tanto el scrap como la cantidad de patrones a utilizar. Pero, por otro lado, también se visualiza que la complejidad computacional que ello conlleva aumenta el tiempo de ejecución. Por lo cual, mientras más se amplía el espacio de búsqueda, disminuyen las posibilidades de llegar a un resultado óptimo en un tiempo reducido. Es por ello que, en el análisis del Enfoque 3, en el que se analizan todos los posibles patrones a generar, se obtienen resultados desalentadores. También es interesante observar que el Enfoque 2 v2, en el cual se trabajó para evitar las repeticiones pero que también llegó al tiempo límite brindado para las ejecuciones, obtuvo iguales o mejores resultado que cualquiera de los modelos aquí planteados.

Si el objetivo fuera una implementación en tiempo reducido para la aplicación en la industria, se podría considerar continuar la exploración del modelo del Enfoque 1. Dicho modelo cuenta con un desempeño interesante y su tiempo de ejecución es sobresaliente en relación al resto. El algoritmo presentado permitió la linealización del modelo original, transformando la variable que asignaba los patrones en un parámetro. Por otro lado, si se desea incorporar mayor realismo en la resolución de la problemática, se debe abordar y profundizar el Enfoque 3. En este se comienzan a incorporar elementos que repercuten en la conformación de los patrones y que están presentes en la industria.

Es importante mencionar que aquí se analizó un caso con un número bajo de pedidos, a modo de ejemplificación. Sin embargo, se considera que algunos modelos son escalables y capaces de abordar la complejidad que implicaría la incorporación de mayor número de pedidos, así como la modificación de los límites máximos y mínimos requeridos en cada caso o incorporación de mayor numero de variables que afecten a la tarea. Estos aspectos serán abordados en la continuidad del proyecto, al igual que potenciales comparaciones con enfoques similares propuestos en la literatura.

Referencias

1. González, C.A.G., Cabrera, J.P.O., Peña, D.L.B.: Cutting stock problem, classification and approaches. Prospectiva. 15, 112-126 (2017).
2. Dyckhoff, H.: A typology of cutting and packing problems. European Journal of Operational Research. 44, 145-159 (1990).
3. Haessler, R.W., Sweeney, P.: Cutting stock problems and solution procedures. European Journal of Operational Research. 54, 141-150 (1991).
4. Delorme, M., Iori, M., Martello, S.: Bin packing and cutting stock problems: Mathematical models and exact algorithms. European Journal of Operational Research. 255, 1-20 (2016).
5. Gilmore, P.C., Gomory, R.E.: A Linear Programming Approach to the Cutting Stock Problem-Part II. Operations Research. 11, 863-888 (1963).
6. Cruz-Reyes, L., Quiroz C., M., Alvim, A.C.F., Fraire Huacuja, H., J., Gómez, S.C., Torres-Jiménez, J.: Heurísticas de agrupación híbridas eficientes para el problema de empacado de objetos en contenedores. Computación y Sistemas. 16, 349-360 (2012).
7. GAMS página principal, https://www.gams.com/, último acceso 18/03/2024.