

Una combinación de un algoritmo voraz con algoritmos genéticos para optimizar la producción de cartón ondulado

TORNADIJO RODRÍGUEZ, TOMÁS F.

Departamento de Informática

Cartonajes Vir, S.A.

Correo electrónico: tornadijo@telecable.es

RESUMEN

En este trabajo se propone la utilización de un algoritmo genético para la optimización del corte continuo de planchas de cartón, un problema habitual en la industria cartonera, donde la minimización de mermas de materia prima y el cumplimiento de los plazos de fabricación son dos objetivos prioritarios de la planificación de la producción.

Las soluciones aportadas por un algoritmo voraz se utilizan como semillas para el algoritmo genético. Se utiliza el operador de recombinación de cruce en un punto.

Palabras clave: industria cartonera; cajas de cartón; algoritmos genéticos; SPP-CP.

Clasificación JEL: C61; L23.

2000MSC: 68T20; 90C27; 90C59.

A Combination of a Greedy Algorithm and Genetics Algorithms to Optimize the Production of Corrugated Board

ABSTRACT

In this paper, we suggest a genetic algorithm for optimizing the continuous cutting of cardboard plates, a common problem in the corrugated industry, where minimizing wastage of raw materials and meeting production deadlines are two priority objectives of production planning.

The solutions provided by a greedy algorithm are used as seeds for the genetic algorithm. It uses the one-point crossover operator.

Keywords: corrugator manufacturing; cardboard boxes; genetic algorithms; SPP-CP.

JEL classification: C61; L23.

2000MSC: 68T20; 90C27; 90C59.



1. INTRODUCCIÓN

El corte de piezas planas de algún material, aprovechando al máximo la superficie de materia prima disponible, es un conocido problema de optimización que, en diferentes variantes, resulta común a diversos sectores industriales, como el de la industria papelera, el del metal o el del vidrio. Este problema adquiere especial importancia y complejidad en la industria del cartón ondulado, donde la elevada cantidad de pedidos y la rapidez con que acontecen los cambios en las condiciones de producción dificultan la consecución de los objetivos de garantizar el cumplimiento de las fechas de entrega de los productos y de minimizar las pérdidas debidas al recorte del material, mermas que conllevan un elevado coste económico y ecológico.

El proceso de producción del embalaje de cartón ondulado comienza con la fabricación, a partir de bobinas de papel, de una plancha de cartón continuo en una máquina denominada onduladora. De esta plancha se cortan las hojas de acuerdo a las especificaciones de los clientes y, tras ser sometidas a un procedimiento de *converting*, resultan transformadas finalmente en embalaje. Las cuchillas de las onduladoras o bien permiten cortar varias bandas de hojas de las mismas dimensiones, procedimiento denominado *simplex* o, más habitualmente, son capaces de cortar varias bandas de dos clases distintas de planchas (*dúplex*), procedimiento que resulta más flexible que el *simplex* y que permite disminuir el volumen del recorte. En cualquiera de los dos casos hay que tener en cuenta que no resulta posible permutar el ancho y el largo de una hoja pues, por razones estructurales, resulta preciso mantener la dirección de la onda del cartón fijada en el diseño del embalaje.

Aquí analizaremos la planificación asociada al procedimiento *dúplex* ya que, además de ser el sistema más utilizado en la industria, en realidad engloba al *simplex* como un caso particular del mismo. La utilización de un sistema *dúplex* conlleva un importante esfuerzo de planificación de la producción, ya que este método puede ser clasificado, entre los problemas de corte y empaquetado, como un caso particular del corte a guillotina *Strip Packing Problem*, (SPP): el *Strip Packing Problem for Corrugator Production* (SPP-CP), que ya ha sido identificado como NP-completo (Rinaldi y Franz, 2007). Intentaremos construir un algoritmo sencillo, que sea capaz de obtener buenas combinaciones y con la suficiente flexibilidad y rapidez como para integrarlo en un entorno de fabricación real.

En el SPP, dado un conjunto de rectángulos $P = \{p_1, \dots, p_n\}$, donde $p_i = (a_i, l_i)$ y una plancha $W = (w, \infty)$, se trata de empaquetar los rectángulos de P en W , minimizando el tamaño del empaquetamiento en la dirección de W . Aunque el problema resulta muy parecido al de las plantas de cartón, el SPP-CP presenta peculiaridades propias, como la fabricación por bandas, y con prioridades particulares, como la necesidad de cumplimentar todos los pedidos asignados, tanto por razones de fechas de entrega del producto, como para respetar los tiempos de la programación de transformación.

1.1. Esquema general

En la Sección 2 describiremos el problema a tratar, esbozando la complejidad combinatoria del mismo, mencionando algunos métodos exactos y heurísticos propuestos para su resolución. En la Sección 3 se explica el concepto de un AG, la literatura sobre su aplicación a nuestro problema y, finalmente, la descripción de un AG diseñado para resolverlo. En la Sección 4 se realizan varias pruebas con diferentes versiones del algoritmo para determinar los mejores operadores. En la Sección 5 se ofrecen las conclusiones y propuestas.

2. DESCRIPCIÓN DEL PROBLEMA

En la Figura 1 se muestra un ejemplo, supuesta la plancha de cartón saliendo del tren ondulator hacia la derecha. Hay tres pedidos: p_1, p_2, p_3 , con i, j, k hojas.

Tenemos dos patrones de corte (combinaciones distintas de uno o dos pedidos con diferentes bandas): el primero, de longitud a completa el pedido p_2 , con una banda, con parte del pedido p_1 , también con una banda. El pedido p_1 se completa con el segundo patrón, de longitud b , con el pedido p_3 , fabricado en dos bandas.

En la Figura 2 hay otro ejemplo donde se representan tres patrones de corte con anchos de bobina diferentes, para combinar tres pedidos hipotéticos de 5, 15 y 16 planchas, en diferentes bandas.

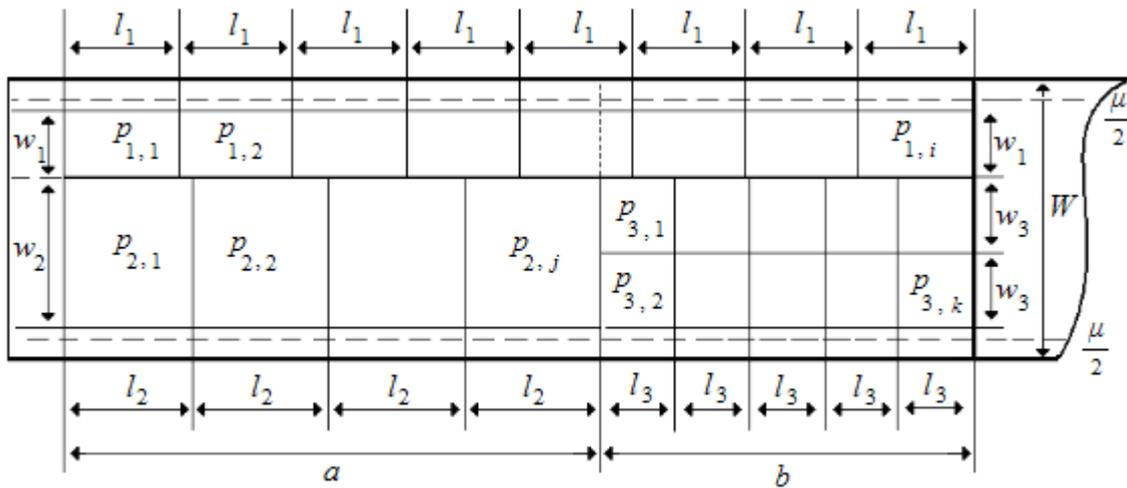


Figura 1. Ejemplo de combinación de pedidos.

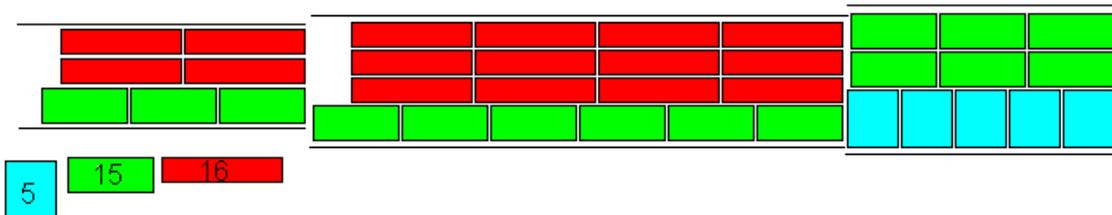


Figura 2. Tres patrones de corte para combinar tres pedidos.

2.1. Formulación del problema

Supuesto un conjunto de n pedidos, $P = \{p_1, p_2, \dots, p_n\}$, para cada p_i , $i = 1, 2, \dots, n$, tenemos los siguientes valores:

- h_i : número de hojas del pedido;
- W_i : ancho del pedido;
- L_i : largo del pedido.

El problema consiste en disponer los rectángulos de los distintos pedidos del conjunto P dentro de rectángulos de la forma $R = W\alpha \times [0, \lambda]$, siendo $W\alpha$ un elemento de $W = \{W1, W2, \dots, Wt\}$ (conjunto de anchos disponibles de bobinas de papel), $0 \leq \lambda$, de forma que se optimice la producción, maximizando los metros cuadrados producidos y minimizando el recorte.

Dos pedidos $p_i, p_j \in P$, $i, j = 1, 2, \dots, n$ (en aquellos casos en los que $i = j$ nos estaremos refiriendo a un único pedido), podrán ser dispuestos sobre un rectángulo R siempre y cuando verifiquen las siguientes restricciones:

- No se permiten las rotaciones ortogonales de las hojas: debido a razones estructurales no resulta posible variar la dirección de la onda del cartón fijada en el diseño del embalaje.
- Los pedidos se colocan en bandas dispuestas paralelamente hasta un máximo de β bandas. Esto quiere decir que a cada pedido $p_i \in P$, $i = 1, 2, \dots, n$, le podremos asignar un número de bandas $b_i \in \{1, 2, \dots, \beta\}$, con la restricción de que si dos pedidos distintos $p_i, p_j \in P$, $i, j = 1, 2, \dots, n$, $i \neq j$, están distribuidos en bandas paralelas, entonces: $b_i + b_j \leq \beta$.
- Existen unas restricciones de recorte lateral: hay un mínimo permitido μ_{\min} y un máximo μ_{\max} . Esto implica que el ancho de la combinación:

$$\text{WComb}_{i,j} = \begin{cases} w_i b_i & \text{si } i = j \\ w_i b_i + w_j b_j & \text{si } i \neq j \end{cases}$$

deberá verificar: $W\alpha - \mu_{\max} \leq \text{WComb}_{i,j} \leq W\alpha - \mu_{\min}$, para todos $i, j = 1, 2, \dots, n$.

- Un patrón de corte tendrá, como máximo, la longitud de su pedido más corto:

$$\text{LComb}_{i,j} = \begin{cases} \frac{l_i \cdot h_i}{b_i} & \text{si } i = j \\ \min \left\{ \frac{l_i \cdot h_i}{b_i}, \frac{l_j \cdot h_j}{b_j} \right\} & \text{si } i \neq j \end{cases} \quad [1]$$

para todo $i, j = 1, 2, \dots, n$; de tal manera que el pedido de mayor longitud deberá completarse en otras combinaciones.

- No se puede fabricar una combinación de uno o dos pedidos por debajo de una longitud mínima de fabricación L_{\min} ; es decir, existe un valor L_{\min} de fabricación tal que $L_{\min} \leq L_{\text{comb}_{ij}}$, para todos $i, j = 1, 2, \dots, n$.

Además, resultarán preferibles aquellas soluciones que resuelvan el problema con menor número de patrones. Existen otras restricciones, como la de número máximo de hendidos, que no vamos a considerar aquí.

2.2 Cálculo del número de combinaciones

Vamos a considerar un conjunto de cuatro pedidos distintos $P = \{p_1, p_2, p_3, p_4\}$ para ilustrar el problema.

2.2.1 Formas de combinar pedidos

En este caso podemos realizar cuatro combinaciones formadas por un único pedido:

$$\{p_1, p_2, p_3, p_4\}$$

y seis combinaciones formadas por pares de pedidos diferentes:

$$\{(p_1, p_2), (p_1, p_3), (p_1, p_4), (p_2, p_3), (p_2, p_4), (p_3, p_4)\}.$$

En total podemos hacer $4 + 6 = 10$ combinaciones de pedidos.

Generalizando para un conjunto de n pedidos, $P = \{p_1, p_2, \dots, p_n\}$, el número de combinaciones que podemos formar con dichos pedidos será:

$$n + C_{n,2} = n + \binom{n}{2}$$

donde $C_{n,2}$ denota el conjunto de combinaciones sin repetición de orden 2 de n elementos.

Aunque en la práctica sí tiene importancia el orden de colocación de un pedido, pues resulta preferible que un pedido mantenga el mismo orden en dos combinaciones consecutivas, esta circunstancia puede modificarse *a posteriori*, una vez que tengamos la lista de combinaciones a fabricar, sin más que intercambiar la posición de los pedidos de la pareja.

2.2.2 Formas de asignar bandas

Para el caso de un único pedido, p_i , $i = 1, 2, \dots, n$, el número de bandas a asignar será cualquier valor $b_i \in \{1, 2, \dots, \beta\}$, lo que supone un total de β posibles asignaciones. En el caso de pedidos distintos, hay que tener en cuenta que ahora sí importa el orden de asignación de bandas a cada pedido de la pareja. En nuestro ejemplo, a una pareja de pedidos distintos (p_i, p_j) , $i \neq j$, con $\beta = 7$, podremos asignarle unas bandas (b_i, b_j) con alguna de las siguientes posibilidades:

$$(b_i, b_j) \in \left\{ \begin{array}{ccccccc} (1,1) & (2,1) & (3,1) & (4,1) & (5,1) & (6,1) & (7,1) \\ (1,2) & (2,2) & (3,2) & (4,2) & (5,2) & (6,2) & (7,2) \\ (1,3) & (2,3) & (3,3) & (4,3) & (5,3) & (6,3) & (7,3) \\ (1,4) & (2,4) & (3,4) & (4,4) & (5,4) & (6,4) & (7,4) \\ (1,5) & (2,5) & (3,5) & (4,5) & (5,5) & (6,5) & (7,5) \\ (1,6) & (2,6) & (3,6) & (4,6) & (5,6) & (6,6) & (7,6) \\ (1,7) & (2,7) & (3,7) & (4,7) & (5,7) & (6,7) & (7,7) \end{array} \right\}$$

Hay un total de β^2 posibles asignaciones de las cuales hay que descartar aquellos casos en los que $b_i + b_j > \beta$, casos que se han representado en negrita y cuyo número se corresponde con la suma de los β primeros números naturales. Es decir, el número de asignaciones posibles de bandas a parejas de pedidos distintos será:

$$\beta^2 - (1 + 2 + 3 + \dots + \beta) = \beta^2 - \frac{\beta(\beta + 1)}{2} = \frac{\beta(\beta - 1)}{2}.$$

2.2.3 Número de patrones de corte

Siendo t el cardinal del conjunto de anchos de máquina, el número de combinaciones de un pedido o dos pedidos distintos (patrones de corte), teniendo en cuenta las asignaciones de bandas y de anchos de bobina, será:

$$c = \left(n\beta + C_{n,2} \left(\frac{\beta(\beta - 1)}{2} \right) \right) t;$$

para $n = 20, t = 4, \beta = 7$:

$$c = \left[20 \cdot 7 + \binom{20}{2} \left(\frac{7(7 - 1)}{2} \right) \right] 4 = 16.520.$$

Si bien éste es un número de combinaciones que puede ser fácilmente explorado por cualquier algoritmo exhaustivo para obtener un conjunto de patrones de corte válidos, el número total de combinaciones que podemos formar con ellos para optimizar la fabricación resulta muy elevado.

2.2.4 Número total de combinaciones

El número real de patrones de corte, una vez aplicadas las restricciones de recorte y longitud es substancialmente menor; por ejemplo, para $n = 12, t = 4, \beta = 7$, la media es de unos 125 patrones, mientras que para $n = 20$ la media estaría en unos 450.

A la hora de combinar, podremos utilizar cada patrón, no solo con su longitud máxima (que viene dada por la fórmula [1]), sino con una determinada fracción de ésta, siempre que la nueva longitud no sea inferior a L_{\min} . El objetivo de este fraccionamiento es conseguir más planchas de cartón que nos faciliten la combinación de otros pedidos.

Supuesto un conjunto inicial de 12 pedidos, solo para realizar la primera combinación podremos elegir entre $125 \times 20 = 2.500$ posibilidades, si probamos todos porcentajes entre el

5% y el 100% de la longitud máxima de esa combinación (y si todos estos porcentajes continúan produciendo una longitud válida). Es obvio que si seguimos concatenando patrones hasta completar todos los pedidos o bien hasta un número máximo aceptable de patrones de corte γ , el número final de combinaciones será elevadísimo.

Algunos autores (Amadeh y Hellström, 2008:6) han propuesto modelos de programación lineal para planificar las ondulatoras, advirtiendo que este método tiene dificultades para resolver instancias de 20 pedidos en un tiempo límite de una hora, decantándose finalmente por trabajar sobre un algoritmo genético (AG) convencional para resolver el problema.

Hay más experiencias al respecto (Jaramillo y Correa, 2008) que concluyen que los métodos basados en AG obtienen mejores resultados que los basados en programación lineal, que no manejan todos los patrones de corte, utilizando una selección de los mismos (Jaramillo y Correa, 2008:6). En general, el rendimiento de estos métodos disminuye rápidamente a medida que crece el tamaño del espacio de búsqueda, con la dificultad de que no resulta posible obtener resultados intermedios en el acercamiento al óptimo, aunque pueden utilizarse como semilladores para acelerar la convergencia de un algoritmo genético (Jaramillo *et al.*, 2009:2).

3. ALGORITMO EVOLUTIVO

3.1. Introducción

Los algoritmos genéticos (AG) son unos métodos heurísticos de búsqueda y optimización, basados en la teoría de la evolución de las especies de Darwin, la cual establece que, en una especie, los individuos de una población mejor adaptados al medio sobreviven y se reproducen con mayor probabilidad, transmitiendo a la progenie sus rasgos adaptativos. Cada individuo de una población porta en sus cromosomas la información genética con la que expresa su grado adaptación al medio, mensurable como *eficacia biológica* o *fitness* (aptitud).

3.2. Descripción de un algoritmo evolutivo

Los algoritmos genéticos, introducidos por Holland (Holland, 1975), intentan emular la acción de la selección natural estableciendo una población inicial compuesta por un conjunto de individuos, cada uno de los cuales contiene una solución inicial (generalmente aleatoria) a un problema determinado codificada en una cadena (cromosoma). Esta población inicial es sometida a unas operaciones que resultan análogas a los mecanismos de perpetuación y evolución que se dan en las poblaciones de las especies naturales:

- Selección. Los cromosomas con mejor aptitud son elegidos con mayor probabilidad para la siguiente generación
- Cruce. Opera sobre dos elementos de la población, preferentemente de aptitud elevada y consiste en combinar las características de sus cromosomas para obtener un nuevo individuo.

- Mutación. Supone modificar aleatoriamente el cromosoma de un individuo, para así incrementar la diversidad genética de la población.

El pseudocódigo de un algoritmo genético es como sigue:

```

{Generar aleatoriamente la población inicial,  $P_i$ 
  Evaluar aptitud  $P_i$ 
  Repetir hasta  $g$  generaciones o aptitud adecuada
    {Seleccionar los individuos de  $P_i$  que pasarán a  $P_{i+1}$ 
      Cruzar o mutar individuos de  $P_i$  y colocar en  $P_{i+1}$ 
       $P_i = P_{i+1}$ 
      Evaluar aptitud  $P_i$ }}

```

Si el algoritmo está bien diseñado, la población converge hacia la mejor solución, que estará codificada en el cromosoma del individuo mejor adaptado de la última generación.

3.3 Algunos ejemplos en la literatura

Resulta muy abundante en las publicaciones sobre el SPP la descripción de algoritmos genéticos de distintas tipologías para resolver el problema (*vid. vgr:* De la Fuente y Gómez, 1999), por lo que parece razonable aplicar estos métodos a la planificación de las ondulatoras.

Se han propuesto diversos tipos de algoritmos genéticos para la resolución del SPP-CP, por ejemplo Damir Vuk (Vuk, 2008) describe un algoritmo genético no convencional, en el que los individuos son combinaciones de uno o dos pedidos (patrones). En cada nueva generación se selecciona el mejor individuo para formar parte de la población final, que constituirá la secuencia de combinaciones a fabricar.

En otro artículo (Velásquez *et al.*, 2007), se describe un AG que utiliza una representación con dos cromosomas: uno que codifica la secuencia de patrones y otro que representa la proporción utilizada sobre la máxima longitud de un patrón (máxima longitud que viene dada por la primera de las ordenes del patrón de corte en completar su número de unidades). Así, la solución vendrá dada por los cromosomas con la secuencia ordenada de patrones y con la longitud de cada uno de ellos. Utiliza una función de compartición o *sharing*, que mide la similitud de un individuo con respecto al resto de la población, para obtener una magnitud con la que se divide la aptitud de cada solución, con el objeto de incrementar la diversidad genética.

Otros investigadores (Amadeh y Hellström, 2008), proponen un algoritmo genético en el que, como en el caso anterior, la recombinación es de cruce de un punto para los cromosomas de los patrones y de cruce uniforme para los cromosomas con las longitudes, aunque los autores llaman la atención sobre la necesidad de estudios comparativos sobre la eficiencia de los diferentes tipos de cruce y mutación.

También es posible utilizar un AG con un único cromosoma, el de las longitudes, para representar el problema (Jaramillo y Correa, 2008:8). Éste será el método que utilizaremos en este ensayo, combinado con un *algoritmo voraz*, que obtendrá una primera solución del problema, solución que agregaremos a los individuos de la población inicial con objeto de acelerar la convergencia del AG.

3.4. Codificación

Un cromosoma se codificará con una cadena de caracteres, de manera que cada gen queda representado por un carácter, cuyo valor ASCII indicará el porcentaje sobre la longitud máxima del patrón de corte que vamos a fabricar (*Vid. vgr: Velásquez et al., 2007:82; Amadeh y Hellström, 2008:7*). Podrá tomar valores entre 0 y 20, de forma que un valor 0 implicará que ese patrón de corte no forma parte de la solución, en tanto que un valor 20 indicará que se utiliza el 100% de la longitud.

3.5. Función objetivo

La mejor solución será aquella que complete todos los pedidos que necesitamos fabricar, fijando como segundo objetivo la minimización del desperdicio de cartón (Vuk, 2008:1), adicionalmente podemos considerar preferible fabricar los pedidos más grandes, en vez de los más pequeños, si no resultase posible combinarlos todos. También hay que tener en cuenta que son preferibles las soluciones con un menor número de patrones, puesto que cada cambio supone un tiempo de inactividad de la máquina.

De acuerdo con estos criterios para medir el grado de adaptación de un cromosoma x_i , de una población de tamaño t , donde $i = 1, 2 \dots t$, podemos utilizar la sencilla función de la Tabla 1.

$f(x_i) = 10^5 \cdot Nped(x_i) + Sped(x_i) - Srec(x_i) - Npat(x_i) \cdot Sn$	
$Nped$	Número de pedidos completados
$Sped$	Superficie de los pedidos
$Srec$	Superficie del recorte
$Npat$	Nº de patrones en la solución
Sn	Área fabricable durante un período equivalente al tiempo medio de cambio.
$\forall Nped(x_i) < 10^2$	
$\forall Sped(x_i) < 10^5$	
$\forall Srec(x_i) < 10^5$	

Tabla 1. Función de aptitud.

Si establecemos en 99.999 m² la máxima superficie de pedidos combinable, conseguiremos que la función de aptitud valore prioritariamente el número de pedidos completados multiplicando dicho número por 10⁵, para que tome un orden de magnitud superior al valor aportado por la superficie. De esta forma, entre varias soluciones que consigan

completar todos los pedidos, la única diferencia vendrá dada por la superficie del recorte: cuanto menor sea, mejores valores tomará la función de aptitud.

A la hora de emparejar los pedidos de un determinado patrón de corte, se puede tolerar un incremento de hasta el 5% en la cantidad de alguno de los pedidos, si ello basta para combinarlo. En tal caso, el valor de superficie enviado a la función de aptitud será el correspondiente a la cantidad original del pedido, mientras que la superficie del recorte será la real; de esta forma, el valor de la función de aptitud será mejor para el caso del patrón de corte que consiga combinar el pedido sin necesidad de incrementar la cantidad del mismo.

Adicionalmente, se detrae del valor de aptitud el área media de cartón que se hubiese podido producir durante el tiempo muerto de una transición de patrón, como si fuese un recorte, con el objetivo de controlar el número de cambios.

3.6. Población inicial

Como paso previo al procedimiento, se obtienen todos los patrones, descartando aquellos que no cumplan las restricciones de recorte lateral y longitud (Amadeh y Hellström, 2008:7), y se almacenan en un vector. Los cromosomas serán cadenas con tantos genes como número de patrones de corte hayamos generado, representando el porcentaje a utilizar sobre la longitud máxima del patrón. Inicialmente, generaremos aleatoriamente el conjunto de porcentajes que formarán parte del genotipo de un cromosoma. Repetiremos esta operación hasta producir todos los individuos de la población inicial.

También se preparó una implementación del *algoritmo voraz* para sembrar la población inicial con patrones de corte productivos. El procedimiento consta de los siguientes pasos:

1. $t = t + 1$.
2. Se toma el patrón de corte t .
3. Se examinan todos los patrones no utilizados y se selecciona aquel que proporcione un mejor valor de aptitud.
4. Si t es menor que el número de patrones y no se han completado todos los pedidos, volver al punto 1.
5. Se calcula el porcentaje de longitud de cada patrón con respecto al máximo y se componen los genes de un cromosoma.

Aunque los patrones se seleccionan con la máxima la longitud factible, perdiéndose posibilidades de combinación, esto también tiene la ventaja de disminuir el número de cambios en la solución final, lográndose con frecuencia resultados perfectamente aceptables.

3.7. Selección

Dado que se ha empleado con éxito la selección por torneo en las experiencias descritas sobre este problema (Velásquez *et al.*, 2007:86; Jaramillo *et al.*, 2009:2), se ha decidido utilizar una

selección de este tipo, por torneo binario determinístico, procedimiento que consiste en tomar aleatoriamente dos individuos de la población y escoger entre ellos aquel individuo con mejor valor de aptitud.

Con una determinada probabilidad, algunos elementos de la población no mutan ni se cruzan, sino que pasan directamente a la siguiente generación. Además se utiliza el elitismo, de manera que se asegura que el individuo mejor adaptado de una generación pase siempre a la siguiente.

3.8. Mutación

A un cierto porcentaje de los individuos seleccionados se les aplica una operación de mutación simple; para ello, se elige al azar un gen del cromosoma y se cambia por el carácter ASCII correspondiente a número generado aleatoriamente entre 0 y 20.

3.9 Cruce

Se han ensayado dos tipos de cruce: el cruce uniforme y el cruce de un punto, por ser los métodos de recombinación más utilizados en la literatura sobre este problema (Velásquez *et al.*, 2007:87; Vuk, 2008:8). En la Figura 3 se muestra un ejemplo del procedimiento. En el caso del cruce uniforme, se crea un solo descendiente a partir de genes tomados de un progenitor u otro, de acuerdo a una máscara de cruce binaria generada aleatoriamente.

El cruce de un punto consiste en cortar dos cromosomas por un punto seleccionado al azar para generar dos segmentos: la cabeza y la cola. Ambos segmentos se unirán para producir un nuevo individuo.

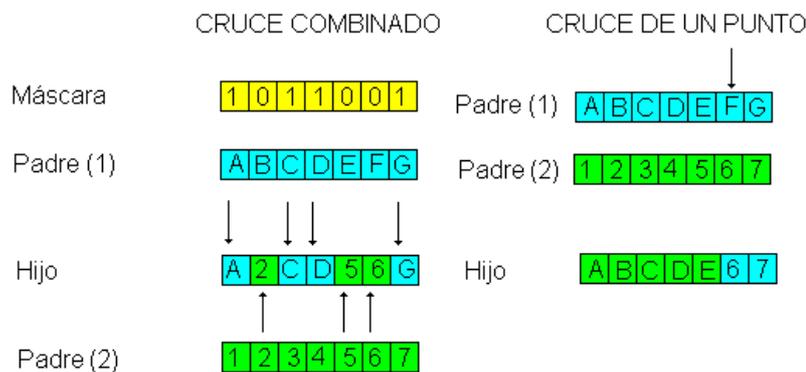


Figura 3. Tipos de recombinación.

4. RESULTADOS

El AG se codificó en Visual Basic, integrándolo con el software de producción, de manera que la selección de las órdenes y su combinación por el AG fuese un proceso ágil, interactivo y reversible. El experimento se realizó con un conjunto representativo de 12 pedidos del mismo grupo o calidad de cartón, efectuándose 20 ejecuciones por cada generación y para cada uno de los algoritmos evaluados (Tabla 3) con el fin de obtener las medias de variabilidad y desempeño. La parametrización de los AG se muestra en la Tabla 2.

PARÁMETROS DE LOS AG							
Población	Nº máximo de generaciones	Probabilidad de cruce	Probabilidad de mutación	γ (Nº máximo de patrones en un cromosoma)	n (Nº de pedidos)	t (Nº de anchos de bobina)	β (Nº máximo de bandas)
300	28	0,97	0,02	12	12	4	7

Tabla 2. Parámetros de los AG.

Generaciones	CRUCE UNIFORME SEMILLADO		CRUCE DE UN PUNTO SEMILLADO		CRUCE UNIFORME		CRUCE DE UN PUNTO	
	Aptitud media	Desviación estándar	Aptitud media	Desviación estándar	Aptitud media	Desviación estándar	Aptitud media	Desviación estándar
1	1.110.677	0	1.110.677	0	847.357	70.121	837.032	67.689
4	1.110.677	0	1.110.677	0	917.637	31.393	888.540	63.306
7	1.110.677	0	1.120.685	31.647	978.681	48.230	958.697	52.524
10	1.110.677	0	1.140.496	48.012	1.019.124	31.751	1.039.091	48.276
13	1.120.569	31.283	1.180.486	48.173	1.109.279	47.225	1.099.413	31.803
16	1.130.449	41.682	1.190.468	42.047	1.139.496	48.314	1.119.120	56.906
19	1.130.493	41.777	1.210.492	244	1.159.632	52.679	1.149.596	51.755
22	1.189.840	41.723	1.210.569	228	1.199.660	31.624	1.179.668	48.370
25	1.209.886	90	1.210.552	199	1.209.747	108	1.189.774	42.151
28	1.200.035	31.397	1.210.601	92	1.209.803	146	1.209.802	162

Tabla 3. Desempeño de los AG.

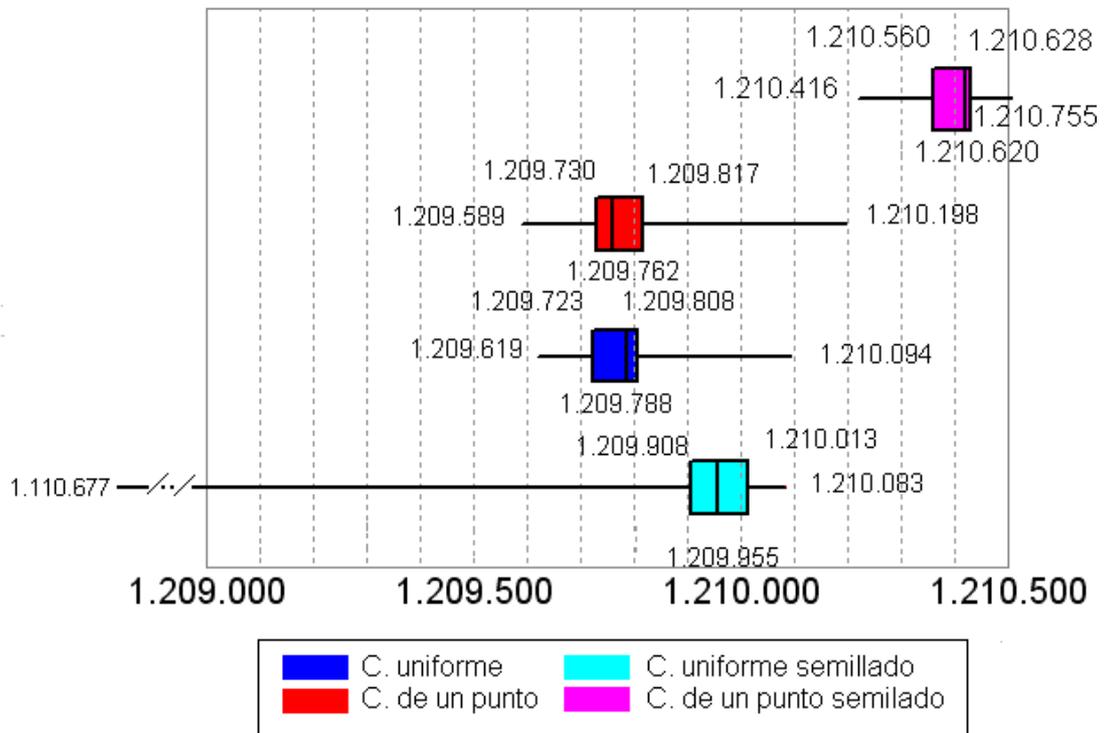


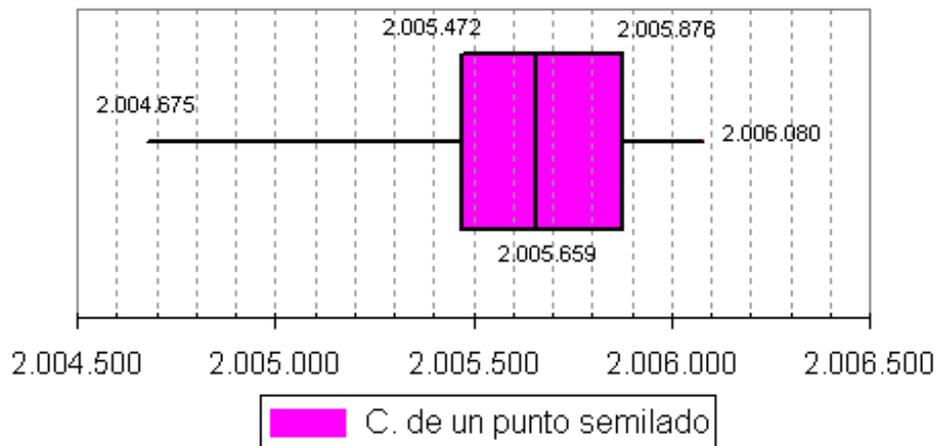
Figura 4. Aptitud en la generación 28ª.

PARÁMETROS DEL AG							
Población	Nº máximo de generaciones	Probabilidad de cruce	Probabilidad de mutación	γ (Nº máximo de patrones en un cromosoma)	n (Nº de pedidos)	t (Nº de anchos de bobina)	β (Nº máximo de bandas)
500	46	0,97	0,02	20	20	4	7

Tabla 4. Parámetros del AG.

Generaciones	CRUCE DE UN PUNTO SEMILLADO	
	Aptitud media	Desviación estándar
1	1.624.106	62.920
6	1.754.595	70.207
11	1.864.668	51.496
16	1.894.645	31.381
21	1.924.945	41.879
26	1.974.775	48.313
31	1.975.291	48.210
36	2.005.146	363
41	2.005.555	420
46	2.005.551	457

Tabla 5. Desempeño del AG.



	Mínimo	1 ^{er} cuartil	Mediana	3 ^{er} cuartil	Máximo
	2.004.675	2.005.472	2.005.659	2.005.876	2.006.080

Figura 5. Aptitud en la generación 46^a.

El tiempo medio de proceso por ejecución para el número más alto de generaciones no llegó a los 5 segundos, en un PC provisto de un procesador Pentium a 2 Ghz. Podemos ver en la gráfica de la Figura 4 que los algoritmos sin semillado convergen hacia una solución que completa los 12 pedidos, con resultados similares para los dos operadores, y cómo la variabilidad disminuye hacia las últimas generaciones.

La implementación con semillado y operador de cruce en un punto parte de una posición de ventaja, desde una solución con 11 pedidos completos, por lo que converge con mayor

rapidez, pues tras 7 generaciones sin superar el aporte del semillado, inicia una elevada variabilidad y comienza a resolver totalmente la cartera de pedidos hacia la generación 19. Por el contrario, el cruce uniforme se estanca durante más tiempo: al principio y entre las generaciones 16 y 22, por lo que no supone una ventaja clara con respecto a las versiones sin semillado.

Este estancamiento inicial, que en menor medida también se aprecia en la versión anterior, se debe a que durante las primeras generaciones la población está dominada por el individuo procedente del semillado. Una posible solución pasaría por introducir más diversidad en la población, por ejemplo incluyendo los resultados generados por el algoritmo voraz a partir de algunas permutaciones del conjunto de patrones; sin embargo, dado que los resultados son lo suficientemente buenos se ha optado por el procedimiento más sencillo.

Una vez comprobada la eficiencia del cruce en un punto, se decidió aplicarlo a una instancia más grande del problema: un conjunto de 20 pedidos de la misma calidad de cartón (Tablas 4 y 5).

El semillado aportó a la población inicial una solución con 16 pedidos completos, facilitando una rápida convergencia del AG, que llegó a valores próximos al óptimo en torno a la generación 36, agotando regularmente los pedidos, sin dejar restos o colas de combinación y con escaso recorte de cartón (Figura 5).

5. CONCLUSIONES

Este trabajo se ha centrado en la búsqueda de una solución, que resultase funcional y operativa en un entorno real, al problema del SSP-CP por medio de un algoritmo evolutivo. De este trabajo podemos extraer las dos siguientes conclusiones:

1. El AG propuesto resulta útil para obtener secuencias de fabricación válidas en un entorno de fabricación real.
2. La utilización del operador de cruce de un punto y el empleo del semillado voraz permite incrementar la velocidad de convergencia del AG.

Tras esta primera aproximación, se ve la necesidad de mejorar la función de aptitud, con objeto de recoger de un modo más preciso todas las variables que intervienen en el proceso, incluyendo no solo los parámetros inherentes a la onduladora, sino también algunos de los relativos al proceso posterior de transformación del cartón.

El enfoque más obvio está basado en el orden de Pareto, donde el valor de aptitud de cada individuo depende, no de cada una de las variables, sino de su eficiencia o dominación dentro de la población. Así el AG, en cada generación, buscará los individuos que no estén dominados por ningún otro, asignándoles el valor más alto. Repitiendo el proceso con los individuos restantes, el AG les asignará magnitudes decrecientes de aptitud, hasta valorar completamente la población.

REFERENCIAS

1. Álvarez-Valdés, R.; Parreño F.; Tamarit J.M. “Reactive GRASP for the strip-packing problem”, *Computers & Operations Research*, vol.35, nº4, pp.1065–1083, 2008.
2. Amadeh, A.; Hellström, D. “The problem of corrugator sequencing and its impact on packaging processes”, 2008: <http://www.lu.se/o.o.i.s?id=12683&postid=1300725>
3. Bortfeldt, A. “A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces”, *European Journal of Operational Research*, vol.172, nº4, pp.814–837, 2006.
4. De la Fuente García, D.; Gómez Gómez, A. “Resolución de problemas de corte de piezas rectangulares mediante algoritmos genéticos”, En: *La gestión de la diversidad: XIII Congreso Nacional, IX Congreso Hispano-Francés, Logroño (La Rioja), 16-18 junio 1999* (coord. Juan Carlos Ayala Calvo), vol.1, 1999, pp.1299–1306, 1999.
5. Holland, J. H. “Adaptation in Natural and Artificial Systems”, *Ann Arbor: The University of Michigan Press*, 1975.
6. Jaramillo, J.D.; Correa, F.J. “Programación lineal y algoritmos genéticos para la solución del problema de corte”, enviado a revisión a la *Revista Colombiana de Computación*, 2008: http://ingenieria-matematica.eafit.edu.co/biblioteca/practicas_investigativas/prac_inv_programacion_lineal_ag_corte.pdf
7. Jaramillo, J.D.; Correa, F.J.; Jaramillo, R. “Desarrollo de un método basado en algoritmos genéticos y programación lineal para la solución de un problema de corte unidimensional”, enviado a revisión a la *Revista de Ingeniería de la Universidad Eafit*, 2009: <http://serverlab.unab.edu.co:8080/wikimedia/memorias/shortpapers/161.pdf>
8. Kureichik V.M.; Miagkikh V.V.; Topchy A.P. “Genetic Algorithm for Solution of the Traveling Salesman Problem with New Features against Premature Convergence”, *TSURE Journal of Engineering*, nº3, 1997.
9. Lodi, A.; Martello, S.; Monaci, M. “Two-dimensional packing problems: A survey”, *European Journal of Operational Research*, vol.141, nº2, pp.241–252, 2002.
10. Martello, S.; Monaci, M.; Vigo, D. “An exact approach to the strip packing problem”, *Technical Paper OR/00/18, Dipartimento di Elettronica, Informatica e Sistemica, Università di Bologna*, 2000.
11. Rinaldi, F.; Franz, A. “A two-dimensional strip cutting problem with sequencing constraint”, *European Journal of Operational Research*, vol.183, nº3, pp.1371–1384, 2007.
12. Velásquez, G.A.; Bellini, G.; Paternina-Arboleda, C. “A multi-objective approach based on soft computing techniques for production scheduling in corrugator manufacturing plants”, *Ingeniería y Desarrollo* 2007, (enero-junio): <http://redalyc.uaemex.mx/redalyc/src/inicio/ArtPdfRed.jsp?iCve=85202106>
13. Vuk, D. “An Approach to the Guillotine Strip Packing Problem”, 2008: <http://www.ceciis.foi.hr/app/index.php/ceciis/2008/paper/view/97/69>