

Título

Dalianys Pérez Pereira

Grupo C411

A.UNO@LAB.MATCOM.UH.CU

Dayany Alfaro González

Grupo C411

A.DOS@LAB.MATCOM.UH.CU

Gilberto González Rodríguez

Grupo C411

A.DOS@LAB.MATCOM.UH.CU

Antonio Jesús Otaño Barrera

Grupo C411

A.DOS@LAB.MATCOM.UH.CU

Resumen

El Resumen en Español debe constar de 100 a 200 palabras y presentar de forma clara y concisa el contenido fundamental del artículo.

Abstract

The English Abstract must have have 100 to 200 words, and present in a clear and concise form the essentials of the article content.

Palabras Clave: Separadas, Por, Comas.

Tema: Tema, Subtema.

1. Introducción

El problema de patrones de corte de piezas rectangulares pertenece a la familia de problemas de corte y empaquetamiento y sus aplicaciones se pueden observar en industrias de perfiles metálicos, corte de maderas, papel, plástico o vidrio en donde los componentes rectangulares tienen que ser cortados de grandes hojas de material. Para estas industrias es de gran importancia realizar este proceso de corte de una manera eficiente buscando minimizar el desperdicio y los demás costos asociados al proceso, teniendo en cuenta las restricciones técnicas y de demanda.

El problema de patrones de corte es un problema de gran complejidad tanto por las características y variables que involucra como por las técnicas que se utilizan para abordarlo, es una temática en constante evolución y muchos investigadores han desarrollado diversos modelos para resolverlo. El interés en este problema puede ser sustentado por su aplicación práctica y el reto que representa pues, en general, es computacionalmente difícil de resolver ya que es un problema de tipo NP-completo, dado que los patrones de empaquetamiento incrementan exponencialmente con el número de rectángulos que deben ser empaquetados.

Este trabajo tiene como objetivo resolver un problema de corte en dos dimensiones, asociado a la industria del papel, presente en una empresa ubicada en la provincia Pinar del Río, Cuba. Se propone el diseño e implementación de un algoritmo que permita determinar qué patrones de corte deben usarse para cortar un conjunto de hojas de forma que se satisfaga una demanda (de hojas más pequeñas) solicitada por el usuario de forma que el desperdicio resultante de los

cortes sea el menor posible. En este caso se permite la rotación de las piezas a colocar y se requiere que los cortes sean de tipo guillotina, es decir, que el corte vaya de un extremo a otro del rectángulo original.

[Texto explicando lo que aborda cada sección]

2. Antecedentes y Enfoques de Solución

El Problema de patrones de corte (CSP, por sus siglas en inglés) fue formulado por primera vez en 1939 por el economista ruso Kantorovich.

Han surgido numerosas investigaciones que abordan diferentes problemas según el tipo de dimensión (1D y 2D) y desde diversos enfoques tales como los métodos exactos, heurísticos y meta heurísticos, pero aún no existe un método global establecido para dar solución a este tipo de problemas, debido a la complejidad asociada.

2.1 Programación Lineal Entera

Casi todos los procedimientos basados en la programación lineal para resolver el problema de patrones de corte se remontan a Gilmore y Gomory, [1], para lo cual, proponen la relajación de la restricción de integridad para la solución de problemas de programación lineal logrando minimizar el desperdicio a través de la generación de columnas evitando el conocimiento explícito o enumeración de todos los patrones desde el principio, ya que bajo este esquema las columnas (patrones) son generadas cuando se requieran [2]. La idea consiste en utilizar el método simplex revisado para resolver el problema de la entrada del patrón de corte

siguiente a la base mediante la resolución de un problema de la mochila asociado. Este método es denominado en la literatura como *delayed column generation technique*, y permite resolver este tipo de problemas en un tiempo computacional mucho menor [3].

2.2 Procedimientos Heurísticos Secuenciales

Los procedimientos heurísticos secuenciales pertenecen a la clase de heurísticas de búsqueda local. La solución se construye mediante la generación de patrones uno a uno hasta que todos los requerimientos de demanda se hayan satisfecho, donde los patrones inicialmente seleccionados deben tener un nivel de desperdicio bajo, un nivel de utilización alto y dejar una serie de requerimientos para poder combinar bien los patrones futuros, evitando así incurrir posteriormente en desperdicios excesivos [3]. La ventaja principal de este método es que puede controlar otros factores aparte del desperdicio y elimina el problema del redondeo al trabajar sólo con valores enteros.

2.3 Procedimientos Heurísticos Híbridos

Este procedimiento consiste en combinar los dos procedimientos descritos anteriormente, de tal forma que se utilice el procedimiento heurístico secuencial para generar una solución, la cual es guardada y utilizada como base inicial en el procedimiento de programación lineal. Posteriormente, el desperdicio es reducido si es posible a través iteraciones adicionales, tal y como lo realiza [4].

Independientemente de la forma como se combinen estos dos métodos, lo más importante del éxito de la unión entre el procedimiento heurístico secuencial y el redondeo de problemas de programación lineal es la selección del criterio apropiado para resolver el problema [5].

2.4 Metaheurísticas

Ante el problema que presenta la búsqueda local y las heurísticas constructivas de quedar atrapadas en óptimos locales, surgen las metaheurísticas a mediados de 1970 pues tienen la capacidad de guiar la búsqueda local para que se escape de los óptimos locales. Muchos de estos algoritmos se han utilizado para resolver el problema de patrones de corte, entre los cuales se destaca *Tabu Search* (TS), *Greedy Randomized Adaptive Search Procedure* (GRASP) [6], Algoritmos genéticos [7, 8] y *Ant Colony Optimization* (ACO) [9, 10], entre otros algoritmos evolucionarios [11].

3. Definición del Problema

4. Propuesta de Solución

5. Desarrollo

En esta sección (o secciones) incluya el contenido fundamental del artículo. No es necesario tener una sección nombrada *Desarrollo*, por el contrario, nombre

las secciones según el contenido que tratan.

5.1 Organización del Documento

Puede agregar secciones y subsecciones según sea necesario para organizar de manera más coherente su artículo. Tenga en cuenta que un documento más plano es más fácil de navegar y entender, pero las subsecciones relacionadas deberían estar agrupadas en una sección común.

Los nombres de las secciones deben ir en mayúsculas, excepto para las preposiciones, conjunciones, y otros vocablos auxiliares.

Empiece un nuevo párrafo cada vez que vaya a comenzar una idea nueva.

5.2 Listas y Descripciones

Para producir listas enumeradas, use el siguiente estilo:

1. Primer Elemento
2. Segundo Elemento
 - (a) Segundo Elemento - Subitem Uno
 - (b) Segundo Elemento - Subitem Dos

Para producir descripciones, use el siguiente estilo:

Primer Elemento con su respectiva descripción.

Segundo Elemento también con su respectiva descripción.

5.3 Figuras

Para producir cuerpos flotantes (figuras ó tablas), asegúrese de numerar y etiquetar correctamente cada figura. Las referencias a las figuras deben estar también correctamente etiquetadas. Por ejemplo, en la Fig. 1 se muestra. . .

Aquí va el contenido de la figura . . .

Figure 1: Figura de ejemplo

5.4 Código Fuente

Para producir código fuente, envuélvalo en una figura flotante y etiquételo correctamente. Por ejemplo, en la Fig. 2 se muestra un código bastante conocido. . .

```
int main(int argc, char** argv)
{
    // Imprimiendo "Hola Mundo".
    printf("Hello, World");
}
```

Figure 2: Código fuente de ejemplo.

5.5 Referencias

Las referencias deben estar agrupadas en una sección al final del artículo, y las citas numeradas correctamente, por ejemplo [12] ó [13]. Incluya toda la información importante de cada referencia, incluídos autor, título, y notas de la edición. En caso de citar sitios web, además de la URL, incluya la fecha en que fue consultado, como en [14].

6. Conclusiones

En esta sección puede incluir las conclusiones de su investigación y las ideas sobre la continuidad del trabajo, en el caso que aplique.

7. Recomendaciones

En esta sección puede incluir recomendaciones sobre posibles formas de continuar la investigación u otros temas relacionados.

References

- [1] P. Gilmore, and R. Gomory, "A linear programming approach to the Cutting Stock Problem-Part II," *Operations Research*, 11(6), 863-888, 1963.
- [2] H. Hideki, and M.J. Pinto, "An integrated cutting stock and sequencing problem," *European Journal of Operational Research* (183), 1353-1370, 2007.
- [3] J. Karelaiti, "Solving the cutting stock problem in the steel industry". Department of Engineering Physics and Mathematics. Helsinki University of Technology, 2-5, 2002.
- [4] Cui, Y.-P., Tang, T.-B. Parallelized sequential value correction procedure for the one-dimensional cutting stock problem with multiple stock lengths. *Engineering Optimization*, 46 (10), 1352-1368, 2014.
- [5] R. Haessler, and P. Sweeney, "Cutting stock problems and solution procedures," *European Journal of Operational Research*, 54, 141-150, 1991.
- [6] MirHassani, S.A., Jalaiean Bashirzadeh, A. A GRASP meta-heuristic for two-dimensional irregular cutting stock problem. *International Journal of Advanced Manufacturing Technology*, 81 (1-4), 455- 464, 2105.
- [7] Wenshu, L., Dan, M., Jinzhuo, W. Study on cutting stock optimization for decayed wood board based on genetic algorithm. *Open Automation and Control Systems Journal*, 7 (1), 284-289, 2015.
- [8] Lu, H.-C.a , Huang, Y.-H.b. An efficient genetic algorithm with a corner space algorithm for a cutting stock problem in the TFT-LCD industry. *European Journal of Operational Research*, 246 (1), 51-65, 2015.
- [9] Lu, Q., Zhou, X. GPU parallel ant colony algorithm for the dynamic one-dimensional cutting stock problem based on the on-line detection. *Yi Qi Yi Biao Xue Bao/ Chinese Journal of Scientific Instrument*, 36 (8), pp. 1774-1782, 2015.
- [10] Díaz, D., Valledor, P., Areces, P., Rodil, J., Suárez, M. An ACO Algorithm to Solve an Extended Cutting Stock Problem for Scrap Minimization in a Bar Mill. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8667, 13-24, 2014.
- [11] Ben Lagha, G.a , Dahmani, N.b , Krichen, S.a. Particle swarm optimization approach for resolving the cutting stock problem. 2014 International Conference on Advanced Logistics and Transport, 2014.
- [12] Donald E. Knuth. *The Art of Computer Programming*. Volume 1: Fundamental Algorithms (3rd edition), 1997. Addison-Wesley Professional.
- [13] Kurt Gödel. *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I*. Monatshefte für Mathematik und Physik 38.
- [14] Wikipedia. URL: <http://en.wikipedia.org>. Consultado en November 10, 2020.