Proyecto 2 Flujo en Redes

Nicolás Hernández1, Nicolás Vásquez2

1 Universidad de los Andes  
C/Tulipan s/n, 28922, Madrid, Spain  
abraham.duarte@urjc.es

2 Universidad de los Andes  
[n.vasquez10@uniandes.edu.co](mailto:n.vasquez10@uniandes.edu.co)

nm.hernandez10@uniandes.edu.co

Abstract

Este documento contiene el desarrollo, explicación, implementación y los resultados del modelo matemático formulado para encontrar el patrón de corte que brinde el menor tiempo posible y cuya implementación obtenga un tiempo computacional bajo. Se realizó una implementación de la formulación de un problema CPP (Chinese Postman Problem) modificado para adecuarlo al problema de cortes utilizando el optimizador Gurobi con el API de Java.

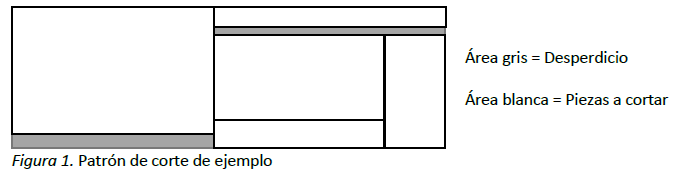
# Introducción

Este documento contiene el desarrollo, explicación, implementación y los resultados del modelo matemático formulado para encontrar el patrón de corte que brinde el menor tiempo posible y cuya implementación obtenga un tiempo computacional bajo. Se realizó una implementación de la formulación de un problema CPP (Chinese Postman Problem) modificado para adecuarlo al problema de cortes utilizando el optimizador Gurobi con el API de Java.

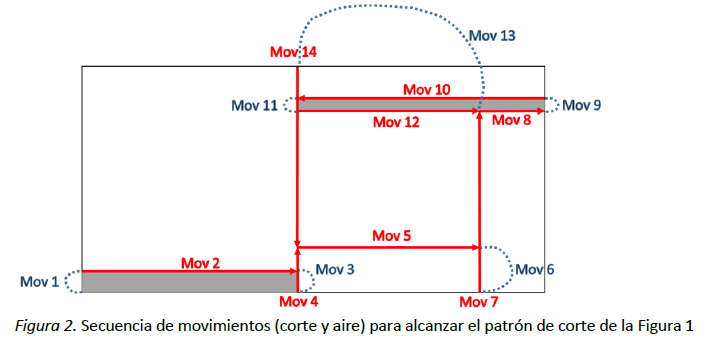
The MIC/MAEB 2017 proceedings will be composed of the electronic manuscripts submitted by the authors and accepted by the Program Committee after the reviewing process. The file format must be PDF (*Portable Document Format*) on an A4 format paper (21cm × 29.7cm).

## Descripción del problema

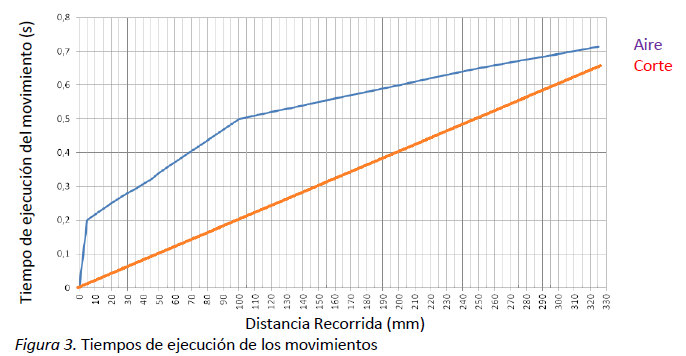
Implementación, experimentación, análisis de resultados obtenidos y elaboración de informe de una metodología de optimización para el problema de secuenciamiento de corte bidimensional de material.



Existen diferentes tipos de movimientos dependiendo de la tecnología de corte, en el problema de corte simulado existen solo dos tipos: movimiento de corte y movimiento en el aire. El primero consiste en un movimiento durante el cual el efector final (marcador) viaja dejando huella en la placa de material (mesa de trabajo). Por otro lado, el movimiento en el aire sucede cuando el cortador (brazo robótico) se desplaza de un lugar a otro sin dejar huella (ver Figura 2).



Ambos movimientos tienen un consumo de tiempo diferente, en especial ambos comportamientos son no lineales, para facilidad de calcular los tiempos de ejecución de los movimientos asuma los tiempos ilustrados en la Figura 3.



El objetivo de problema de secuenciamiento de corte es minimizar el tiempo total para llevar a cabo la tarea de corte, el tiempo total está compuesto por la suma de los tiempos de los movimientos más el tiempo de cómputo requerido por la metodología de solución.

Cabe resaltar que cortes contiguos entre piezas se pueden realizar una única vez, mientras que cortes contiguos con los límites de la placa no son necesarios de realizarlos (ver Figura 2).

La estructura del Proyecto 1 se espera que conserve un diseño metodológico (en formato artículo, anexo en este anuncio), donde se especifique claramente la descripción del problema, el esquema de optimización (modelo matemático, pseudocódigo, etc.), junto a las ventajas y desventajas de este. Además de esto, se debe implementar la metodología (en cualquier lenguaje de programación computacional o matemático) y se debe realizar la experimentación y validación del desarrollo, a través del uso de los patrones de corte obtenidos por (Cuellar-Usaquén, 2018) para las 17 instancias (GCUT1-17) propuestas por (Beasley, 1985) contenidas en el archivo comprimido PatronesCorte.rar (adjunto en este anuncio). Sumado a la experimentación se debe adicionar un análisis de los resultados obtenidos con relación a las soluciones alcanzadas.

## Esquema de optimización

Para resolver este problema, se va a realizar un grafo, donde los nodos representen las esquinas de las figuras que se cortan y los arcos el movimiento de la máquina (ya sea en corte o sin realizar corte). Luego, para resolver el problema, se formuló la solución como un problema de optimización lineal entera, donde se utilice la estructura de un modelo clásico de flujo en redes solucionando un problema de ruta más corta, añadiendo una restricción que obligue a que se visiten los arcos que corten las piezas y que solamente se pase por un corte una sola vez.

# Metodología

## Realización del grafo

Para realizar el grafo, se carga el archivo de texto que contiene las coordenadas de las figuras a cortar. En primer lugar, todas estas esquinas son un nodo en el grafo, así que se agregan todos los nuevos nodos sin alguna. Luego, para todo nodo existente se crea un nodo “gemelo” que represente la misma esquina en la misma coordenada. Esto, se hace con el motivo de representar que hay dos maneras de llegar a una esquina: una es por aire y otra es cortando. Adicionalmente, se crean dos nodos cuyo objetivo es ser el nodo inicial y nodo final para el problema de “ruta más corta” y no representan ninguna coordenada.

Luego, se crean los arcos entre los nodos. Primero, debe haber un arco entre todos los nodos de aire, puesto que existe un movimiento entre todas las esquinas del patrón que se realice por aire. Además, el costo vinculado con este arco será igual al tiempo que tome entre las coordenadas del nodo incial y del nodo final viajando por aire de acuerdo con los tiempos de la Figura 3. Por otro lado, todos los nodos corte, tienen un arco que los conecta con la esquina más cercana a la cual se pueda llegar por medio de un corte factible, es decir, sin cortar por espacios donde se arruinen las figuras. Así mismo, el costo vinculado a este arco, será el tiempo que tome moverse entre ambas esquinas haciendo un corte. Por lo tanto, este arco representa realizar un corte entre las dos coordenadas (inicial y final) del arco.

Adicionalmente, se crean arcos con un costo vinculado de 0, dado que es una transición entre dos nodos que no implica un movimiento en coordenadas de la máquina de corte. Estos arcos, son los que conectan cada nodo de corte con su respectivo nodo de aire, dado que esto no representa ningún movimiento para la máquina, sino una transición necesaria para identificar si va a realizar un corte o si se va a mover por el aire. Además, existe un arco entre el nodo inicio y todos los nodos que no sean el nodo final, así como hay un arco entre todos los nodos y el nodo final excepto por el nodo inicio. Estos arcos tampoco tienen ningún costo (tiempo) asociado. Por último, cabe recalcar que todos los arcos son no dirigidos, y por lo tanto se crean dos arcos dirigidos por cada uno (uno de ida y uno de vuelta).

## Formulación del problema

Se realizó una formulación lineal entera para solucionar el problema de corte como un problema de flujo en redes de ruta más corta, utilizando los nodos inicio y final explicados en la realización del grafo, añadiendo una restricción para que se cortaran los arcos de corte una sola vez y llegar a la solución para cada instancia. La formulación es la siguiente:

**Conjuntos:** Se cuenta con un conjunto de arcos total que representan los movimientos de la máquina de corte, un subconjunto de arcos de corte y un conjunto de nodos que representa las esquinas de las figuras que se deben cortar.

**Variables de decisión:** Se crea una variable de decisión binaria para cada arco en el problema, que toma el valor de 1 si el arco hace parte de la solución y 0 de lo contrario, es decir, cuando el arco se utiliza la máquina se va a mover entre las coordenadas de la esquina y la .

**Parámetros:** Como parámetro se obtiene el tiempo de movimiento para todos los arcos del problema.

**Función objetivo:** Se debe minimizar el tiempo total de corte:

(1)

**Restricciones:**

1. Restricción de balance:

(2.1)

(2.2)

(2.3)

1. Restricción para pasar obligatoriamente una sola vez por los arcos de corte:

(3)

1. Naturaleza de las variables

(4)

# Implementación Java + Gurobi

Se utilizó el optimizador Gurobi para resolver el problema formulado, utilizando el API de Java. La aplicación ejecuta una clase llamada Model.java. Esta, primero carga el archivo de datos del patrón, arma el grafo en la clase Reader.java y genera la matriz de adyacencias. Luego, desde la clase Model.java se construye el modelo a optimizar: se crean las variables, restricciones y el sentido del problema (maximización o minimización). Luego de esto, resuelve el modelo y obtiene los arcos que hacen parte de la solución óptima. Por último, ordena la solución como un solo camino para generar un archivo de texto con las coordenadas que debe visitar la máquina en orden indicando si el movimiento es de corte o por el aire.

# Resultados

## Resultados para las instancias proporcionadas

Para todas las instancias, se genera un camino óptimo ordenado a seguir. Estos resultados se encuentran en los anexos.

## Resultados en tiempo computacional

A continuación, presentamos un cuadro comparativo mostrando el tiempo computacional que demora

# Análisis de resultados

Los resultados presentan el mayor tiempo computacional para

**Gráfica No. 1:** Número de nodos vs. Tiempo procesamiento

**Gráfica No. 2:** Número de nodos vs. Tiempo de corte total

**Gráfica No. 3:** Tiempo total para cada patrón

# Conclusiones

En primer lugar,

# Illustrations and Formulae

## Tables and Figures

Tables and Figures (in short illustrations) should be placed as close as possible to the place where they are first discussed. Illustrations must be numbered sequentially, depending on their type, and their caption should be placed below the illustration and it must read as Figure/Table #: caption text (see Figure 1 and Table 1 for concrete examples).

Illustrations should be horizontally centered in the page. The captions must be centered as well, and they must be typeset in 11pt regular font. Captions should be placed within a vertical space of 0.5cm above and below the text.

| Algorithm | Average cost |
| --- | --- |
| A | **1** |
| B | 2 |
| C | 5 |

Table 1: An example table

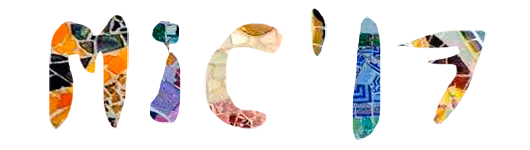


Figure 1: MIC logo

## 

Figure 2: MAEB logo

## Formulae

Mathematical formulae must be centered and numbered, using sequential Arabic numbers in parentheses as in the following example. The text in the formulae must be typeset in 11pt italic font.

(1)

References

1. James Brown. Why the work of John Smith must not be cited. Personal communication, 2011.
2. James Brown and John Smith. How not to cite papers. In John Smith, editor, *Proceedings of the First International Conference on Modern Bibliometrics (MODBIB 2009*), pages 20–30, Pasadena (CA), USA, July 25–28 2009.
3. John Smith. How to make citations. *Journal of Modern Bibliometrics*, 1:1–10, 2010.
4. John Smith. Against the accusations of James Brown. Technical Report 01-11, CS Department, University of Serendipity, 2011.