

Diseño del encofrado para muros usando encofrados modulares

Formwork Layout for Walls Using Modular Formworks

Carlos A. Martínez  ¹, Juan F. Díaz ² y
Robinson Duque ³,

Recibido: 25 de septiembre de 2019
Aceptado: 19 de noviembre de 2019

Cómo citar / How to cite

C. A. Martínez, J. F. Díaz, R. Duque, “Diseño del encofrado para muros usando encofrados modulares”, *TecnoLógicas*, vol. 22, pp. 1-18, 2019. <https://doi.org/10.22430/22565337.1509>



- ¹ MSc. en Ingeniería, Grupo de Investigación AVISPA, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Cali-Colombia, carlos.andres.martinez@correounivalle.edu.co
- ² PhD. en Informática, Grupo de Investigación AVISPA, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Cali-Colombia, juanfco.diaz@correounivalle.edu.co
- ³ PhD en Ingeniería Eléctrica, Grupo de Investigación AVISPA, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Cali-Colombia, robinson.duque@correounivalle.edu.co

Resumen

El concreto u hormigón es uno de los materiales más versátiles usados en la construcción. En la actualidad, se usa extensivamente en la industria, ya que puede ser moldeado de formas casi arbitrarias a través de moldes rígidos denominados encofrados. La manera en que se disponen los distintos elementos para formar el encofrado y las restricciones asociadas, genera retos computacionales interesantes que impactan directamente la industria de la construcción. En este artículo se propone la implementación de un modelo de restricciones que genera encofrados para muros rectilíneos usando formaletas modulares. Este enfoque se basa en la descomposición de muros y su representación en instancias del problema de cambio de monedas. Para este propósito, se utiliza el concepto de descomposición rectangular orientada para simplificar la entrada del problema y abordar la solución como un problema de optimización por restricciones (COP). Los experimentos con entradas de prueba ofrecen soluciones en tiempos inferiores a 200 segundos, cuyo resultado es un enfoque práctico y eficiente para la compleja tarea del diseño de encofrados.

Palabras clave

Diseño de encofrados, Muros rectilíneos, Programación por restricciones, Problema de Optimización por Restricciones, Problema del cambio de Monedas.

Abstract

Concrete is one of the most versatile materials used in construction. At present, it is extensively employed in the industry as it can be molded in almost any shape through rigid molds called formwork. The arrangement of the different elements that compose the formwork and its associated restrictions pose interesting computational challenges that directly impact the construction industry. In this article, we propose the implementation of a model with constraints that generates a formwork layout for rectilinear walls using modular forms. This approach is based on the decomposition of walls and their representation as instances of the change-making problem. For that purpose, we used the concept of oriented rectangular decomposition to simplify the problem input and address the solution as a Constraint Optimization Problem (COP). The experiments with test inputs provided solutions in less than 200 seconds, thus offering a practical and efficient approach to the complex task of formwork design.

Keywords

Formwork Layout, Rectilinear Walls, Constraint Programming, Constraint Optimization Problem, Change-making problem.

1. INTRODUCCIÓN

1.1 Los encofrados modulares

Un encofrado es un sistema de moldes que se usa para dar forma al concreto mientras fragua. Los encofrados pueden ser temporales (se retiran una vez que el concreto ha fraguado) o llegar a ser parte integral de la construcción. Existen distintos tipos de encofrados y varias formas de clasificarlos. Este artículo se enfoca en un tipo particular del encofrado modular para muros en concreto reforzado.

Los encofrados modulares están constituidos por elementos normalizados y prefabricados, que son muy útiles en obras de gran volumen y que tienen secciones de formas repetitivas como, por ejemplo, los edificios de apartamentos.

Los encofrados modulares permiten reducir los costos de la construcción, debido a su versatilidad para adoptar distintas formas, su fácil manejo en la obra y a que son reutilizables.

El principal elemento de un encofrado modular es la formaleta. Una formaleta es una pieza rectangular que se une con otras para formar los moldes para el vaciado del concreto. Cuando no es posible usar una formaleta, se ponen uno o más *fillers* de ajuste que en su funcionalidad asemejan formaletas: tienen el mismo largo, pero se fabrican con un ancho menor que el de la formaleta más pequeña. Aunque hay muchas más piezas (esquineros, herrajes variados, entre otros), la complejidad del diseño del encofrado radica en cómo se disponen las formaletas y los *fillers* de ajuste para crear el molde de una estructura en concreto.

1.2 El diseño del encofrado y la programación por restricciones

Un diseño de encofrado especifica la forma como deben armarse los elementos que lo componen. Al realizar un diseño de encofrado se buscan los siguientes

objetivos: minimizar la cantidad de formaletas requeridas y minimizar el área no cubierta por elementos normalizados.

La dificultad del diseño del encofrado modular radica en que la cantidad disponible de tamaños de formaletas convierte el problema del diseño en uno de tipo combinatorio, que puede asemejarse al problema de cambio de monedas [1] o al problema de empaquetamiento bidimensional [2].

Adicionalmente, las formas no rectilíneas para las áreas a modular (aquellas en las que se deben colocar formaletas), contribuyen a que el problema sea complejo de tratar por la dificultad inherente en determinar cuál es la mejor forma de dividir dichas áreas de modulación para hacer la entrada del problema más manejable.

Tal descomposición llegó a ser considerada un problema NP-completo (o problema que puede ser resuelto en tiempo polinomial con una máquina de Turing no determinista), en [3] se propone un algoritmo eficiente para esta labor.

En este artículo se propone un modelo para muros rectilíneos y se plantea el uso de una descomposición rectangular orientada verticalmente para dividir el área de modulación en rectángulos simples. Adicionalmente, la solución se aborda utilizando Programación por Restricciones (CP), el cual es un poderoso paradigma de programación que puede ser utilizado para resolver problemas combinatorios.

Típicamente, CP combina algoritmos de búsqueda (*backtracking*) y utiliza técnicas de propagación para filtrar valores inconsistentes en el dominio de las variables para así reducir el espacio de búsqueda [4].

Un problema en CP es representado como un CSP (Problema de Satisfacción de Restricciones), el cual está conformado por un conjunto de variables con sus respectivos dominios y las restricciones asociadas. Dependiendo de la naturaleza

del problema, se puede buscar una solución (o problemas de decisión), todas las soluciones (o problemas de satisfacción de restricciones), o encontrar la solución óptima que satisfaga una función objetivo, como en los problemas de optimización por restricciones (COP) [5].

1.3 El diseño de encofrados en la industria

La eficiencia en el proceso de armado de encofrados es uno de los factores que determinan el éxito de un proyecto de construcción, en términos de velocidad, calidad, costo y seguridad del trabajo.

Por consiguiente, un correcto diseño del encofrado tiene un gran efecto en el proceso entero de construcción [6].

Los costos del encofrado se estiman entre un 40 y 60 % de los costos totales relacionados con el concreto (i.e., costo del concreto, refuerzos, materiales para el encofrado, mano de obra para el armado y vaciado del concreto) y representan cerca del 10 % del total de los costos de construcción [7].

La utilización racional de los recursos naturales disponibles para la construcción, el reciclaje, y la reutilización de materiales que minimice el impacto ambiental, conllevan a un cambio de mentalidad en la industria de la construcción [8]. En este sentido, la reutilización inteligente de los componentes del encofrado y las tecnologías subyacentes para la producción y uso del concreto reforzado, [9] que garanticen diferentes propiedades físico-mecánicas en cuanto a permeabilidad, disminución de gradientes térmicos, humedad, densidad, resistencia a la compresión [10], responden a las características de sostenibilidad para mejorar las condiciones medioambientales y ahorrar recursos.

El diseño de los encofrados para la construcción no debe ser subjetivo y debe ser bien planeado antes de que inicie la obra, dado que un proceso no optimizado de construcción resulta en incremento de

costos. En la industria de la construcción, los principales criterios para la selección de determinado sistema de encofrado son, usualmente, la eficiencia para la construcción, el costo de los componentes y la repetitividad del trabajo [11]. Para el caso de los sistemas basados en encofrados modulares, la reducción de costos puede ser alcanzada disminuyendo los precios de arrendamiento de componentes y mano de obra para el armado del encofrado; sin embargo, múltiples criterios pueden ser estudiados al momento de diseñar un sistema de encofrado [12], [13].

En [12] los autores estudian la influencia de distintos factores de construcción sobre la productividad laboral para el armado de encofrados.

Como resultado, cuantifican el impacto del diseño teniendo en cuenta la variabilidad del tamaño de las columnas, la repetición y su geometría.

Los resultados muestran efectos significativos de estos factores en la productividad laboral, que pueden utilizarse para proporcionar a los diseñadores comentarios sobre qué tan bien elaborados son los diseños, considerando los requisitos del principio de construcción. Este estudio, una vez más, confirma la importancia del proceso de diseño de encofrados que se aborda en la presente investigación.

En [13] se propone el uso redes neuronales para estimar costos de mano de obra en la construcción de encofrados y se propone un modelo que permite ser utilizado por los planeadores de obra para estimación de costos. Sin embargo, a diferencia de este trabajo de investigación, se deja de lado la tarea del diseño de encofrados, la cual tiene también un impacto directo en la complejidad del armado y, por consiguiente, en la cantidad de horas de trabajo requeridas.

Para utilizar un encofrado de forma eficiente, el área de construcción es usualmente dividida en zonas de trabajo.

De igual manera, el proceso de construcción se divide en conjuntos de operaciones más pequeñas para equipos de trabajo especializados que laboran en diferentes zonas de forma paralela.

El tamaño y forma de las zonas de trabajo determinan el máximo reuso que se puede dar a los componentes del encofrado en zonas de trabajo consecutivas, de tamaño y forma similares [14]. De esta manera, contar con un diseño específico permite establecer la cantidad de elementos a utilizar puesto que, si se ordenan demasiados componentes para el encofrado, algunos permanecerán almacenados en desuso. Por otro lado, si se ordenan pocos elementos, entonces la rotación de los componentes incrementa la posibilidad de daños en ellos; adicionalmente, la velocidad y calidad del trabajo también se puede ver afectada por la falta de disponibilidad de ciertos componentes en determinados momentos.

Problemas relacionados han sido estudiados a través del uso de algoritmos genéticos para resolver el problema del diseño del encofrado y de la selección del encofrado, de acuerdo con las propiedades de sus materiales. Tal es el caso de [15] donde los autores abordan el problema del encofrado para losas (techos) con formas no rectilíneas, usando 12 tipos diferentes de formaletas modulares. En el caso de estudio presentado, lograron reducir los costos de arrendamiento de formaletas en un 10.4 % y de las áreas no cubiertas por formaletas estándar en un 11.9 %.

Igualmente, en [16] y [17] los autores proponen el uso de algoritmos genéticos para hacer una adecuada selección del tipo de encofrado a usar, de acuerdo con las características de la construcción y del sistema de encofrado.

En cuanto al uso de CP para resolver los problemas relacionados con el encofrado, en [18] los autores presentan el problema del emparejamiento de encofrados (FPP), probando que este es un problema NP-duro y plantean un método

de solución que relaja restricciones para poder encontrar soluciones óptimas en el 50 % de los casos estudiados; y en [19] los autores abordan otro interesante problema, que es el de determinar cuál es el inventario mínimo de materiales de encofrado que una obra de construcción requiere. Este es otro problema NP-duro.

Problemas asociados también han sido estudiados utilizando programación entera mixta (MIP) para encofrados verticales con el objetivo de reducir costos de arrendamiento [9] y minimizar costos de encofrados en construcciones residenciales [20].

1.4 Estructura del artículo

El resto de este artículo está estructurado así: en la Sección 2 se describe el modelo junto con la metodología para la simplificación de la entrada del problema. En la Sección 3 se presentan los resultados obtenidos de la aproximación planteada. Finalmente, en la Sección 4 se presentan las conclusiones y se plantean las líneas de trabajo futuro.

2. MODELADO DEL PROBLEMA

2.1 Reglas de la colocación de formaletas

En el diseño del encofrado modular, las formaletas y *fillers* de ajuste se colocan de acuerdo con las siguientes reglas:

- Cubren el área de modulación total o parcialmente, de acuerdo con unas tolerancias definidas.
- Se organizan en filas.
- Tienen una orientación estándar (la parte más larga va sobre el eje y).
- Solo pueden rotarse 90°.
- No se solapan.
- Se colocan de forma contigua.
- No sobresalen del área de modulación en los laterales y el borde inferior.

- Pueden sobresalir en la parte superior del área de modulación y solo en el borde superior del muro correspondiente.
- Se colocan las mismas formaletas en ambas caras del muro.

Además, las filas de formaletas obedecen las siguientes reglas:

- Se pueden rotar 90° si tienen cierta altura y es la última fila en la parte superior.
- Si hay una fila sobre otra, la fila superior debe ser de igual o menor altura.
- Hay un máximo de *fillers* de ajuste por fila.
- Si una fila está formada por formaletas en posición rotada no se usan *fillers* de ajuste en esa fila.
- Las filas se componen solo por formaletas en posición estándar o solo por formaletas rotadas.

2.2 El área de modulación

La mayor complejidad del problema radica en la colocación de las formaletas y *fillers* de ajuste. La disposición de los demás elementos del encofrado está sujeta a la cantidad y disposición de las formaletas o a la forma propia de los muros (finales de muro, intersecciones, vanos o huecos de puertas y ventanas). Es así como un área de modulación se puede definir como el área de la superficie de los muros en la cual debe colocarse una formaleta o un *filler* de ajuste. En otras palabras, el área de modulación es el área no cubierta de los muros que queda después de colocar los esquineros para las intersecciones, y los tapamuros para los vanos y finales de muro. La combinatoria involucrada para colocar los tapamuros y los esquineros es baja y puede abordarse con técnicas de programación tradicionales. Para los fines de este trabajo solo se tiene en cuenta el área de modulación.

2.3 Simplificación de la entrada del problema

Los muros en concreto reforzado de una construcción (muros para abreviar) son estructuras tridimensionales.

Los elementos que conforman un encofrado son igualmente objetos tridimensionales. Una de las primeras tareas a realizar para abordar el problema del diseño del encofrado para muros rectilíneos como uno netamente combinatorio, es simplificar las entradas de tal forma que el detalle de la ubicación de las piezas y su forma en el espacio se deje al sistema de visualización o a etapas posteriores de procesamiento.

Las formaletas (y *fillers* de ajuste) pueden modelarse en términos de su largo y ancho, como un rectángulo simple en dos dimensiones, al igual que el área de modulación sin pérdida de información, usando las proyecciones de las caras de los muros y una descomposición rectangular orientada verticalmente.

2.3.1 Descomposición rectangular orientada verticalmente

La descomposición rectangular orientada verticalmente es el proceso mediante el cual se convierte el área de modulación en rectángulos simples sin huecos en su interior. La manera más sencilla de entender la descomposición rectangular orientada verticalmente para un área de modulación es con una imagen de ejemplo (Fig. 1).

Para empezar, se deben obtener las proyecciones de las caras externas de cada muro y con ellas se hace la descomposición rectangular orientada (Fig. 2).

Si trazamos rayos verticales que coincidan con los bordes laterales de cada vano, o donde se produzca un cambio de dirección en los bordes del polígono rectilíneo, cada polígono se descompone en rectángulos simples sin huecos en su

interior. Identificando cada rectángulo de forma única y conociendo la posición de su esquina inferior izquierda, es posible reconstruir el polígono rectilíneo original (Fig. 3).

Siguiendo con el ejemplo, se puede observar que los rectángulos 1 a 4, 6 a 8, 10, 12 y 13 limitan en su parte superior con el borde superior del muro.

Asimismo, los rectángulos 1 a 5 y 8 a 13 tienen contracara (para cada formaleta o *filler* de ajuste colocado en el encofrado, hay otro en la misma posición en la cara opuesta del muro, pero invertido).

Los rectángulos 6 y 7 hacen parte de una intersección de muros en L. Al colocar

un esquinero en el encofrado, cada uno podría dividirse en dos rectángulos más (denominados respectivamente 6.1, 6.2, 7.1 y 7.2) y al menos un par de ellos no tener contracara, porque el esquinero interno cubre parte del área de cada muro (Fig. 4).

La descomposición rectangular orientada verticalmente debe tener esto en cuenta y definir los rectángulos respectivos como rectángulos sin contracara. Así, todas las formas obtenidas siguen siendo rectángulos simples y se pueden manejar de la misma forma. Es así como la descomposición rectangular final queda como se presenta en la Fig. 5.

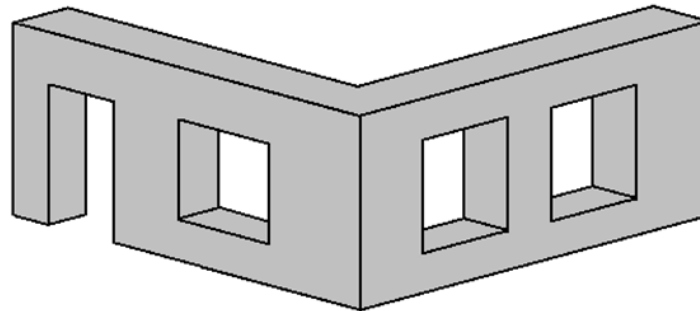


Fig. 1. Ejemplo de dos muros rectilíneos simples en tres dimensiones
Fuente: elaboración propia.

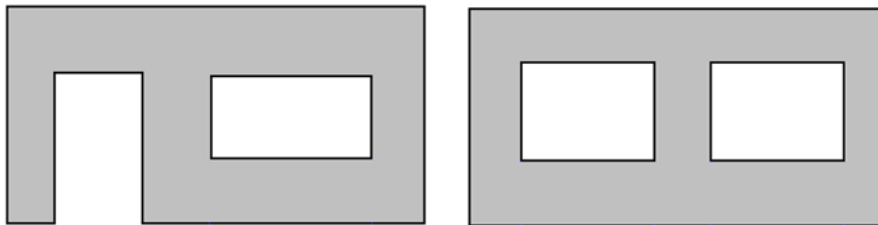


Fig. 2. Proyecciones de las caras externas de ambos muros que generan dos polígonos rectilíneos
Fuente: elaboración propia.

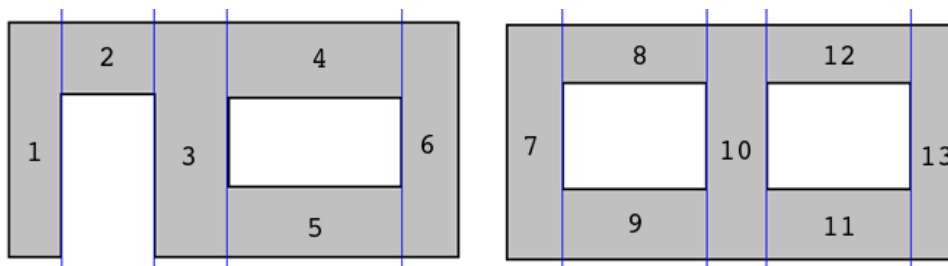


Fig. 3. Descomposición rectangular inicial. Fuente: elaboración propia.

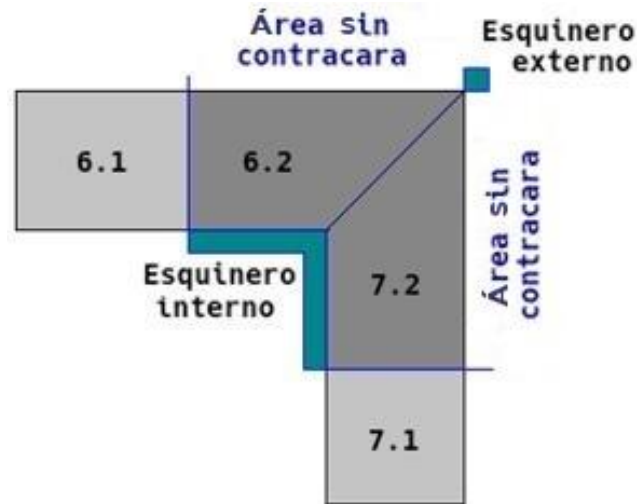


Fig. 4. Vista superior de una intersección de muros en L. 6.2 y 7.2 generan rectángulos que no tienen contracara. Fuente: elaboración propia.

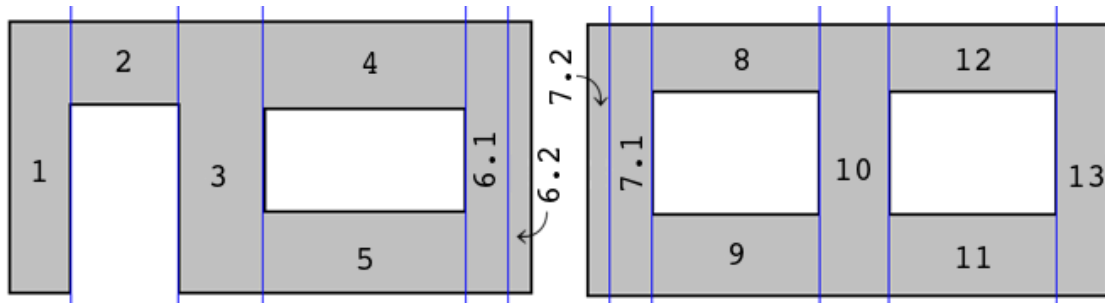


Fig. 5. Descomposición rectangular orientada completa Fuente: elaboración propia.

2.4 Abordando el problema como un problema de cambio de monedas

El problema de cambio de monedas busca encontrar el número mínimo de monedas de determinadas denominaciones, tales que sumen una cantidad específica de dinero. Este problema es un caso particular del problema de la mochila [1].

Cuando abordamos el diseño del encofrado ideal con la simplificación de la entrada planteada en la sección anterior, este puede ser resuelto como un problema de cambio de monedas, ya que tiene las siguientes características:

-Cada rectángulo generado en la descomposición rectangular orientada del área de modulación es un problema independiente. La solución óptima para cualquier conjunto de

rectángulos está formada por el conjunto de soluciones óptimas de cada rectángulo individual.

-Para cada rectángulo el número de filas es independiente del número de columnas. Si se tiene un rectángulo que se va a cubrir con rectángulos más pequeños, manteniendo la alineación al máximo, se termina creando un arreglo de filas y columnas de formaletas. Encontrar la cantidad mínima de filas es independiente de encontrar la cantidad mínima de formaletas que debe ir en cada fila (cantidad de columnas por fila).

Al plantear el modelo del problema de esta manera, se obtienen las siguientes propiedades:

-La salida indica cómo debe ser la solución. El resultado de la modulación indica qué tipo de formaletas y qué

cantidad forman la modulación, pero no indica cómo deben estar ubicadas (coordenadas en el plano de cada formaleta para cubrir el rectángulo). Aun así, es información suficiente para componer la solución teniendo en cuenta que las formaletas se colocan en filas, de forma contigua y sin solapamientos.

-El modelo de restricciones lo constituyen dos Solvers de restricciones independientes. Cada dimensión (largo y ancho) de un rectángulo se trabaja de forma independiente con un Solver que hace la modulación a lo largo (busca el número mínimo de filas de formaletas para cubrir la altura del rectángulo) y otro a lo ancho (busca el número mínimo de formaletas por fila para cubrir el ancho del rectángulo).

-El diseño del encofrado se compone a partir de las salidas de los dos Solvers de restricciones. Se requiere combinar las salidas de ambos Solvers para obtener el encofrado de un área de modulación (Fig. 6)

2.5 Definiciones formales

Definición 1: el rectángulo a modular. sea r un rectángulo de una descomposición rectangular orientada

verticalmente de los muros a modular como en (1):

$$r = (l, a, b, d) \quad (1)$$

$r.l$ representa el largo del rectángulo r o su *offset* en el eje y desde su esquina inferior izquierda.

$r.a$ representa el ancho del rectángulo r o su *offset* en el eje x desde su esquina inferior izquierda.

$r.b$ es una variable booleana que indica si r está ubicado en la parte superior del muro.

$r.d$ es una variable booleana que indica si r pertenece a una sección del muro con contracara.

Definición 2: las formaletas. Las formaletas se describen a través de conjuntos finitos de valores (únicos):

F_{largos} es el conjunto de *offsets* posibles de la formaleta ubicada en la modulación en el eje y . El i -ésimo elemento del conjunto F_{largos} se representa de la forma $F_{largos}.i$.

La misma notación se aplica para el resto de los conjuntos y listas para representar e identificar sus elementos como se muestra en (2):

$$F_{largos} = F_{largos_std} \cup F_{largos_rot} \quad (2)$$

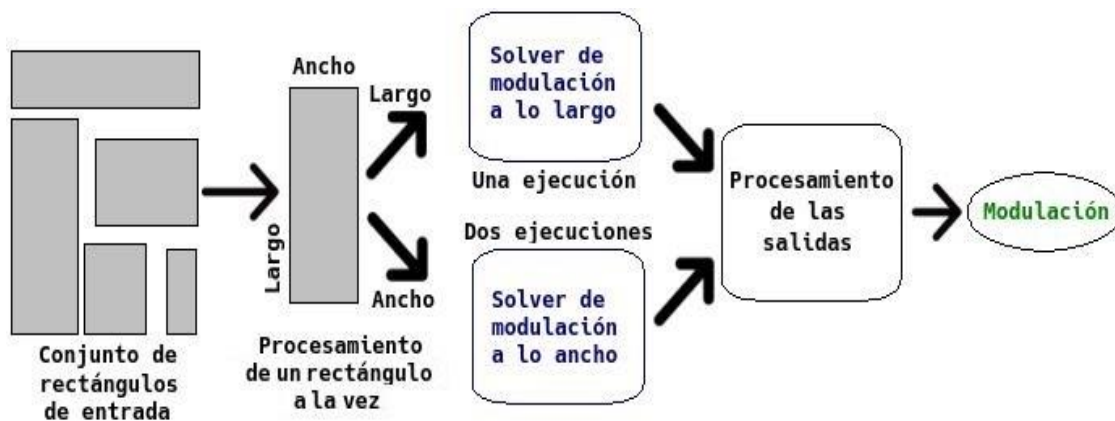


Fig. 6. Diagrama de bloques de la implementación del modelo basado en el problema de cambio de monedas
Fuente: elaboración propia.

F_{largos_std} es el conjunto de largos que pueden tener las formaletas cuando están en posición estándar.

F_{largos_rot} es el conjunto de largos que pueden tener las formaletas cuando están en posición rotada.

F_{anchos} es el conjunto de *offsets* posibles de la formaleta ubicada en la modulación en el eje x . Se tiene que:

$$F_{anchos} = F_{anchos_std} \cup F_{anchos_rot} \cup F_{anchos_fills} \quad (3)$$

$$F_{anchos_fills} \cap F_{anchos_std} = \emptyset \quad (4)$$

En las restricciones (3) y (4), F_{anchos_std} es el conjunto de anchos que pueden tener las formaletas cuando están en posición estándar.

F_{anchos_rot} es el conjunto de anchos que pueden tener las formaletas cuando están en posición rotada.

F_{anchos_fills} es el conjunto de anchos de los *fillers* de ajuste.

Los *offsets* son tomados desde la esquina inferior izquierda de la formaleta, tal y como se ubica en la modulación.

Definición 3: una modulación. dados r , F_{largos} y F_{anchos} se dice que m es una modulación del rectángulo r , es decir, indica cómo cubrirlo en un diseño de encofrado con las tolerancias máximas definidas y su notación se muestra en (5).

$$m = (F_{ILS}, COLS_{std}, COLS_{rot}) \quad (5)$$

F_{ILS} es una lista de enteros mayores o iguales que cero donde $F_{ILS}.i$ representa la cantidad de filas de largo $F_{largos}.i$ que forman la modulación de r a lo largo (sobre el eje y).

$COLS_{std}$ es una lista de enteros mayores o iguales que cero donde $COLS_{std}.i$ representa la cantidad de anchos de formaletas $(F_{anchos_fills} \cup F_{anchos_std}).i$ que forman la modulación de r para una fila de formaletas en posición estándar.

$COLS_{rot}$ es una lista de enteros mayores o iguales que cero donde cada valor $COLS_{rot}.i$ representa la cantidad de anchos de formaletas $F_{anchos_rot}.i$ que forman la modulación de r para una fila de formaletas rotadas.

Para las tres listas de m se introducen las restricciones (6) a (12):

$$card(F_{ILS}) = card(F_{largos}) \quad (6)$$

$$card(COLS_{std}) = card(F_{anchos_fills} \cup F_{anchos_std}) \quad (7)$$

$$card(COLS_{rot}) = card(F_{anchos_rot}) \quad (8)$$

$$F_{ILS} = F_{ILS}_{std} \cup F_{ILS}_{rot} \quad (9)$$

$$COLS_{std} = COLS_{fills} \cup COLS_{fstd} \quad (10)$$

$$card(COLS_{fills}) = card(F_{anchos_fills}) \quad (11)$$

$$card(COLS_{fstd}) = card(F_{anchos_std}) \quad (12)$$

F_{ILS}_{std} representa la cantidad de largos para las filas formadas por formaletas en posición estándar.

F_{ILS}_{rot} representa la cantidad de largos para las filas que necesariamente deben estar rotadas.

$COLS_{fills}$ representa la cantidad de anchos de *fillers* usados.

$COLS_{fstd}$ representa la cantidad de anchos de formaletas usadas en posición estándar.

En el contexto de listas, \cup significa concatenación y $card$ cardinalidad.

Definición 4: largo de la modulación. Representada por l_m es el largo que pueden tener las filas de las formaletas, respetando las tolerancias. Desde la óptica del problema del cambio, l_m representa el valor a devolver y F_{largos} son las denominaciones de monedas con las que se puede formar dicho valor.

Definición 5: ancho de la modulación. representado por a_m es el ancho total de una fila de formaletas, respetando las tolerancias. Desde la óptica del problema del cambio, a_m representa el valor a devolver y F_{anchos} son las denominaciones de monedas con las que se puede formar dicho valor.

Definición 6: las tolerancias. Cierta área de un rectángulo puede quedar sin cubrir por formaletas. Además, si el rectángulo está ubicado en la parte superior de un muro, las formaletas pueden sobresalir de la parte superior del rectángulo hasta determinada altura:

– tv es la tolerancia vertical que indica cuánto espacio vertical puede quedar en la parte superior del rectángulo sin cubrir.

– tve es la tolerancia vertical de exceso que indica el sobrepaso máximo de las formaletas que están en la parte superior si el rectángulo está en la parte superior de un muro.

– tr es la tolerancia de rotación que indica cuándo la fila de la parte superior de la modulación puede ser formada por formaletas en posición rotada.

– th_{std} es la tolerancia horizontal que representa el máximo espacio horizontal no cubierto cuando las formaletas están en posición estándar.

– th_{rot} es la tolerancia horizontal que representa el máximo espacio horizontal no cubierto cuando las formaletas están en posición rotada.

– $max_fillers$ es la cantidad máxima de $fillers$ de ajuste que pueden usarse para formar a_m .

Definición 7: rotación de la última fila. La fila de formaletas que está en la parte superior puede estar compuesta de formaletas rotadas 90° , si su largo es menor o igual que tr .

Definición 8: mejor solución. una modulación m_1 es mejor que una modulación m_2 para un mismo r si cumple

al menos una de las siguientes características:

–La cantidad de formaletas usadas es menor.

–El valor absoluto de la diferencia entre el área de la modulación y el área del rectángulo es menor.

2.6 Restricciones para la modulación a lo largo

La modulación a lo largo, en esencia, toma el largo del rectángulo (valor a cambiar) y con la lista de largos (monedas), obtiene la menor cantidad de filas posibles (cambio) respetando las tolerancias definidas. Este Solver se ejecuta una sola vez para obtener el valor óptimo, formado a partir del menor número de filas posibles y la menor diferencia entre el largo del rectángulo y el largo modulado.

El conjunto de restricciones es el siguiente:

Restricción L1: cantidad de divisiones. Para $card(FILS) = n$, se introducen las restricciones (13) y (14):

$$\forall i \text{ } FILS.i \geq 0 \quad (13)$$

$$divisiones = \sum_{i=1}^n FILS.i \quad (14)$$

El valor $divisiones$ es la cantidad de divisiones o filas del rectángulo a modular.

Restricción L2: largo de la modulación. El largo de la modulación, l_m viene dado por las restricciones (15) y (16):

$$l_m = \sum_{i=1}^n FILS.i * F_{largos}.i \quad (15)$$

$$\delta = |l_m - r.l| \quad (16)$$

δ representa la diferencia entre el largo del rectángulo y el largo modulado.

Restricción L3: cantidad de rotaciones. Solo puede haber, a lo sumo, una fila que obligatoriamente deba estar rotada. Este valor se representa con can_rot para lo cual se introducen las restricciones (17) y (18):

$$0 \leq can_rot \leq 1 \quad (17)$$

$$can_rot = \sum_{i=1}^{card(FILLS_{rot})} FILS_{rot.i} \quad (18)$$

Restricción L4: largo que se puede modular. Si el rectángulo limita con la parte superior del muro, el largo de la modulación l_m puede ser mayor que el largo del rectángulo $r.l$. Si r no está en la parte superior del muro, el largo de la modulación l_m debe ser menor o igual a $r.l$, así como en (19) y (20):

$$r.b = true \Rightarrow (r.l \leq l_m \leq r.l + tve) \quad (19)$$

$$r.b = false \Rightarrow (r.l - tv \leq l_m \leq r.l) \quad (20)$$

Restricción L5: función objetivo. El Solver debe satisfacer las restricciones anteriores sujeto a minimizar el valor δ y el valor $divisiones$ como se muestra en (21):

$$objetivo = v_1 * \delta + v_2 * divisiones \quad (21)$$

v_1 y v_2 son enteros positivos que le asignan peso a cada valor.

2.7 Restricciones para la modulación a lo ancho

La modulación a lo ancho, en esencia, toma el ancho del rectángulo (valor a cambiar) y con la lista de anchos posibles de formaletas (monedas), obtiene la menor cantidad de formaletas posibles (cambio) para cubrirlo respetando las tolerancias definidas.

El conjunto de restricciones es el siguiente:

Restricción A1: cantidad de divisiones. Sea $COLS$ una lista de enteros mayores o iguales que cero tal que $card(COLS) = card(F_{anchos}) = n$, donde cada valor $COLS.i$ representa la cantidad de anchos de formaletas $F_{anchos.i}$ que forman la modulación para una fila. Se introducen las restricciones (22) a (26):

$$COLS = COLS_{std} \cup COLS_{rot} \quad (22)$$

$$\forall i COLS.i \geq 0 \quad (23)$$

$$divisiones = \sum_{i=1}^n COLS.i \quad (24)$$

$$f_rot = false \Rightarrow \forall i COLS_{rot.i} = 0 \quad (25)$$

$$f_rot = true \Rightarrow \forall i COLS_{std.i} = 0 \quad (26)$$

El valor $divisiones$ es la cantidad de divisiones o formaletas de la fila a modular. Las restricciones (25) y (26) garantizan que la fila se componga exclusivamente de formaletas en posición estándar o exclusivamente de formaletas en posición rotada.

Restricción A2: ancho de la modulación. El ancho de la modulación de una fila, a_m , viene dada por las restricciones (27) a (30):

$$a_m = \sum_{i=1}^n COLS.i * F_{anchos.i} \quad (27)$$

$$r.a - t \leq a_m \leq r.a \quad (28)$$

$$t = \begin{cases} th_{std}, & \text{si } f_rot = false \\ th_{rot}, & \text{de lo contrario} \end{cases} \quad (29)$$

$$\delta = r.a - a_m \quad (30)$$

δ representa la diferencia entre la suma de los anchos de las formaletas de la fila y el ancho del rectángulo.

Restricción A3: fillers usados. la variable fu cuenta la cantidad de $fillers$ de ajuste que se usan en la modulación a lo

ancho, así como se denota en las restricciones (31) a (33):

$$fu = \sum_{i=1}^{card(COLS_{fills})} COLS_{fills}.i \quad (31)$$

$$0 \leq fu \leq fum \quad (32)$$

$$fum = \begin{cases} max_fillers, & \text{si } f_rot = false \\ 0, & \text{de lo contrario} \end{cases} \quad (33)$$

Restricción A4: función objetivo. El Solver debe satisfacer las restricciones anteriores y está sujeto a minimizar el valor δ y el valor *divisiones* como en (34):

$$objetivo = v_1 * \delta + v_2 * divisiones \quad (34)$$

v_1 y v_2 son enteros positivos que le asignan peso a cada valor.

2.8 Procesamiento de las salidas de los Solvers

Cada Solver genera parte de la salida que representa el diseño del encofrado o modulación ideal para el área de modulación del problema. Las salidas de ambos Solvers deben combinarse para generar la respuesta esperada que, posteriormente, puede ser representada en un sistema CAD (*Computer-aided design*) o BIM (*Building Information Modelling*). En términos generales, el algoritmo hace lo siguiente:

-Para cada rectángulo r de R y su correspondiente lista $FILS$ de la modulación m obtiene los largos de cada fila, a partir de los valores de largos de formaletas en F_{largos} .

-Para cada fila de $FILS$, conociendo su largo con la correspondiente lista $COLS_{std}$ o $COLS_{rot}$ de la modulación m , determina la cantidad de formaletas y *fillers* de ajuste para formar fila a partir de los valores de anchos de formaletas en F_{anchos} .

-Con la cantidad de formaletas por fila, se hace el conteo de estas, tanto de forma global (total de formaletas y *fillers* de ajuste usados), como de forma individual por tipo.

La salida en texto para un rectángulo r es como se muestra en la Tabla 1.

En la salida para cada rectángulo modulado hay cinco bloques de datos:

-Datos del rectángulo: esta es la información del rectángulo junto con su identificador único.

-Datos de la modulación en altura: especifica cuántas filas de formaletas se requieren para cubrir el largo rectángulo, la altura de cada una de ellas y la altura cubierta.

-Datos de la modulación a lo ancho para las filas de formaletas en posición estándar: especifica cuántas formaletas se requieren para formar la fila, el ancho de cada una de ellas y el ancho cubierto.

-Datos de la modulación a lo ancho para las filas de formaletas en posición rotada: similar al caso anterior, pero para la fila que va rotada porque su altura es menor o igual que la altura de rotación.

-Total de formaletas requeridas: esta es la cantidad de formaletas que se requieren para cubrir todo el rectángulo.

En el ejemplo de la Tabla 1. se tiene un rectángulo con contracara que requiere 20 formaletas (10 por cara) organizados en dos filas. Para formar la fila inferior se necesitan seis formaletas de 2100x600, cuatro de 2100x550, cuatro de 2100x280 y dos de 2100x230. Para la fila superior se requieren dos formaletas de 2100x150 y dos de 1500x150. El rectángulo no queda cubierto en su totalidad y, por lo tanto, se deben usar elementos no estándar para cubrir toda el área faltante. Todas las medidas están dadas en milímetros.

Tabla 1. Ejemplo de salida en texto plano del resultado, las unidades están en milímetros
Fuente: elaboración propia.

```
# datos rectángulo #
Rectangulo_id=1
Altura=2259
Largo=3691
Borde_superior_muro=false
Contracara=true
```

```
#.datos mod_altura.#
Total_filas_modulacion_altura=2
Filas_modulacion= {2100 150}
Altura_modulacion=2250
```

```
#.datos mod_largo_std.#
Total_cols_modulacion_largo_pos_std=8
Columnas_modulacion_largo_pos_std= {600 600 600 550 550 280 280 230}
Largo_modulacion_pos_std=3690
```

```
#.datos mod_largo_rot.#
Total_cols_modulacion_largo_pos_rot=2
Columnas_modulacion_largo_pos_rot= {2100 1500}
Largo_modulacion_pos_rot=3600
```

```
#.total formaletas para el rectángulo.#
Total_formaletas_r=20
```

3. RESULTADOS Y DISCUSIÓN

Para implementar los conjuntos de restricciones planteados se usó MiniZinc [21]. MiniZinc se define a sí mismo como un lenguaje de modelado de restricciones *open source* que permite modelar problemas de satisfacción de restricciones y problemas de optimización en un lenguaje de alto nivel, independiente del lenguaje o librería de CP usado.

MiniZinc permite definir modelos con restricciones y funciones de optimización en un lenguaje de alto nivel que posee gran expresividad. Adicionalmente, compila estos modelos en un archivo FlatZinc, el cual sirve de entrada para una amplia variedad de Solvers que permiten realizar diversas pruebas sin tener que reescribir el modelo bajo los detalles del Solver empleado.

Experimentos con Gecode [22], Cbc [23] y Haifa CSP [24] reportaron que Cbc ofrece mejores tiempos de ejecución para el modelo presentado. Gecode es una librería abierta de C++ para modelar y resolver problemas de restricciones; Haifa CSP es

un Solver para problemas de restricciones de dominios finitos que utiliza algoritmos basados en SAT (problema de satisfacción booleana); Cbc es un Solver de programación entera mixta escrito en C++.

La implementación se evaluó en su desempeño con entradas definidas en milímetros de 10, 100 y 500 rectángulos, generados aleatoriamente, con largos que oscilaban entre 100 mm y los 8 m y anchos que oscilaban entre los 100 mm y los 12 m.

Los intervalos sobre los cuales se generaron los tamaños aleatorios de los rectángulos de entrada se escogieron de tal forma que exceden a los que se encuentran en el mundo real, dado que las alturas de los muros son generalmente 2.4 m o 2.7 m, y el ancho de un muro no excede comúnmente los 5 m. Además, si las dimensiones están dadas en milímetros son, por regla general, múltiplos de 10.

Por ejemplo, en la práctica es poco probable que un rectángulo mida 2413 mm de ancho, aunque sí es probable que mida 2410 mm.

La generación de rectángulos con tamaños que no eran múltiplos de 10 se

hizo con el fin de estresar el modelo planteado y ver qué tan eficiente era ante entradas de gran tamaño y de combinatoria compleja. Se observó que ante entradas en que no eran múltiplos de 10, los Solvers tendían a tardar más tiempo en dar una solución óptima.

De forma intuitiva, se podría pensar que modular un rectángulo de 2413 mm ancho tarda alrededor del mismo tiempo que uno de 2410 mm; sin embargo, ninguna combinación directa de anchos de formaletas y *fillers* de ajuste suma 2413, lo cual conlleva a una exploración mayor.

Como sistema de encofrados modular, se usó el sistema de Acero-Madera de FORSA [25] que fue escogido por la empresa beneficiaria del proyecto asociado a este artículo, dado que el mismo se usa para las soluciones de arrendamiento de encofrados y, además, posee 119 tipos de formaletas y 21 tipos de *fillers* de ajustes distintos, lo que crea un gran espacio combinatorio para crear un diseño de encofrado óptimo.

Todas las pruebas se hicieron en un computador de escritorio con 16 GB en RAM y un procesador Intel Core i5-750 a 2.67 GHz.

En la Tabla 2. se presentan los tiempos de ejecución para las entradas de prueba usando Cbc.

Tabla 2. Tiempos de ejecución con Cbc
Fuente: elaboración propia.

Cant.r	Modelo a lo largo	Modelo a lo ancho	Tiempo total
10	<1s	<1s	<2s
100	11s	22s	33s
500	72s	112s	184s

Como es de esperarse, el aumento en la cantidad de rectángulos de entrada impacta directamente el tiempo de modulación. También se puede observar que la modulación a lo ancho toma casi el doble de tiempo que la modulación a lo largo. Esto se debe principalmente a que la cantidad de anchos de formaletas y *fillers* de ajuste (17 para formaletas y 3 para *fillers*) es mayor que la cantidad de largos (7 en total) y, por lo tanto, generan espacios combinatorios distintos para la búsqueda de soluciones.

Los tiempos de modulación a lo largo para entradas de 60 cm hasta 10 m en incrementos de 1 cm usando Cbc se pueden ver en Fig. 7. En el gráfico se puede observar que los tiempos tienden a aumentar conforme el largo de la modulación es mayor. Asimismo, se observa, al igual que en los siguientes gráficos, que el tiempo de ejecución no depende únicamente del tamaño de entrada del rectángulo. Hay cierta incertidumbre respecto al tiempo de ejecución que es propia de la exploración que realizan los algoritmos de búsqueda y propagación en CP. Por otro lado, los tiempos de modulación a lo ancho para entradas de 60 cm hasta 10 m en incrementos de 1 cm con Cbc se pueden ver en Fig. 8 y Fig. 9.

En los gráficos se observa que, para la modulación de una fila en posición estándar, los tiempos varían poco en función del ancho del rectángulo. Sin embargo, para las filas en posición rotada los tiempos tienden a aumentar en función del aumento del ancho del rectángulo.

Diseño del encofrado para muros usando encofrados modulares

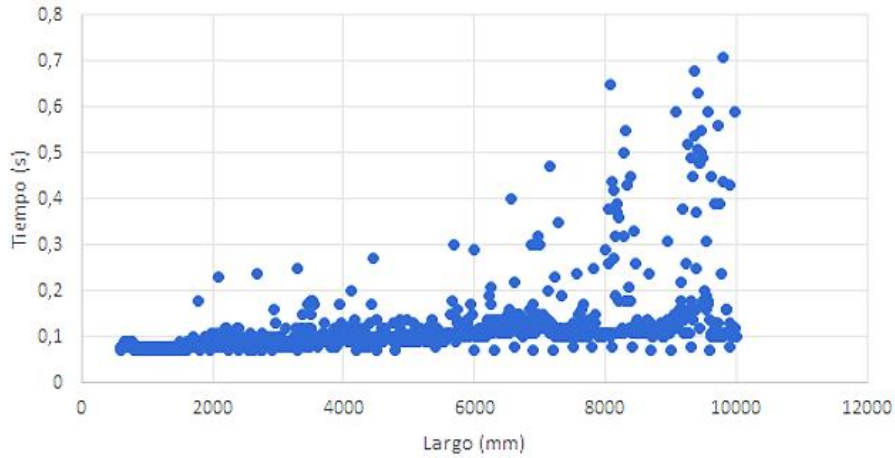


Fig. 7. Tiempos de modulación a lo largo. Fuente: elaboración propia.

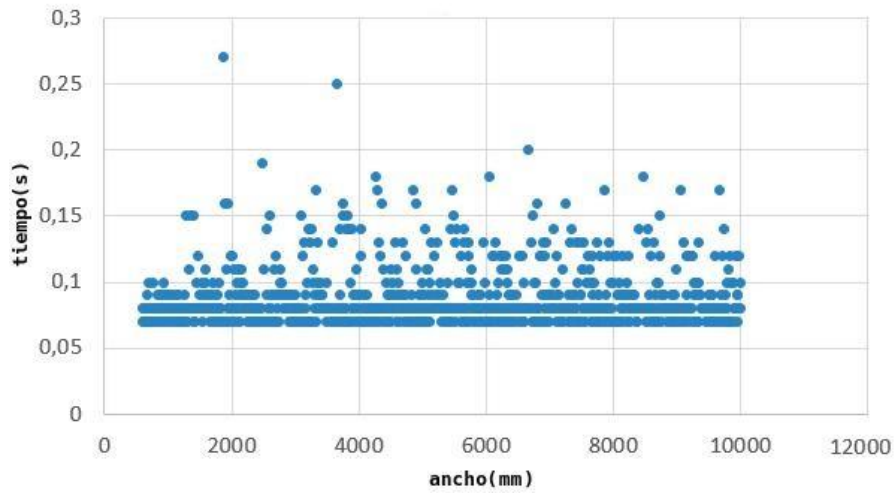


Fig. 8. Tiempos de modulación a lo ancho para una fila en posición estándar
Fuente: elaboración propia.

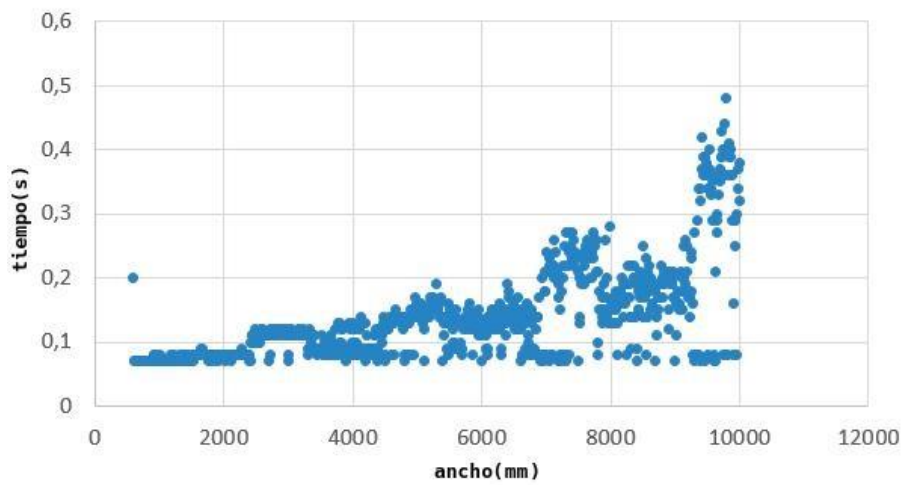


Fig. 9. Tiempos de modulación a lo ancho para una fila en posición rotada
Fuente: elaboración propia.

4. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se abordó el diseño de encofrados para muros *rectilíneos* utilizando formaletas y *fillers* de ajuste. Para este propósito, se propuso una metodología basada en la descomposición rectangular orientada verticalmente para simplificar la entrada del problema. Adicionalmente, se abordó la solución como un problema de optimización por restricciones y el modelo planteado se basó en el problema de cambio de monedas.

El uso de la programación por restricciones demostró ser eficiente en términos del tiempo de ejecución para un conjunto de pruebas realizadas utilizando medidas reales de formaletas y *fillers* y rectángulos de entrada de tamaños arbitrarios.

Son varios los temas sobre los que se puede continuar este trabajo. Desde el punto de vista de la investigación se destacan los siguientes:

-Crear un modelo que genere el mejor encofrado teniendo en cuenta el inventario de formaletas y *fillers* de ajuste disponibles.

-Crear un modelo de restricciones que genere el encofrado para losas (techos).

-Extender los modelos para manejar muros y losas con formas no *rectilíneas* (p. ej. culatas y techos a dos aguas).

Desde el punto de vista del desarrollo aplicado, el trabajo se puede continuar con la implementación de este modelo planteado en un software de encofrado para su uso en producción por parte de la empresa beneficiaria de este proyecto, algo que ya hace parte de un proyecto de i+D+i entre la empresa beneficiaria y la Universidad del Valle.

5. AGRADECIMIENTOS

El proyecto asociado a este artículo fue financiado con recursos del Convenio Interadministrativo de Cooperación de

Ciencia y Tecnología No. CI-1115-2016 para la “Formación e Innovación para el Fortalecimiento de la Competitividad del sector TIC de la Región: FormaTIC e InnovaTIC Valle del Cauca, Occidente” celebrado entre la Corporación PACIFITIC y el Instituto Financiero para el Desarrollo del Valle del Cauca - INFIVALLE, el cual fue financiado con recursos del SGR asignados al proyecto BPIN 2014000100051.

Igualmente, los autores agradecen a FORSA S.A. y a su filial de renta de encofrados, todo el apoyo prestado para la realización de este proyecto.

5. REFERENCIAS

- [1] D. Pearson, “A polynomial-time algorithm for the change-making problem,” *Oper. Res. Lett.*, vol. 33, no. 3, pp. 231–234, May. 2005. <https://doi.org/10.1016/j.orl.2004.06.001>
- [2] A. Caprara y M. Monaci, “Bidimensional packing by bilinear programming,” *Math. Program.*, vol. 118, no. 1, pp. 75–108 Apr. 2009. <https://doi.org/10.1007/s10107-007-0184-7>
- [3] V. Soltan y A. Gorpinevich, “Minimum dissection of a rectilinear polygon with arbitrary holes into rectangles,” *Discrete Comput. Geom.*, vol. 9, no. 1, pp. 57–79, Jan. 1993. <https://doi.org/10.1007/BF02189307>
- [4] F. Rossi, P. van Beek, and T. Walsh, *Handbook of Constraint Programming*, New York: Elsevier, 2006. [https://doi.org/10.1016/S1574-6526\(06\)X8001-X](https://doi.org/10.1016/S1574-6526(06)X8001-X)
- [5] K. Apt, *Principles of Constraint Programming*, Amsterdam: Cambridge University Press, 2003. <https://doi.org/10.1017/CBO9780511615320>
- [6] S. Biruk y P. Jaskowski, “Optimization of Vertical Formwork Layout Plans Using Mixed Integer Linear Programming,” *Int. J. Civ. Eng.*, vol. 15, no. 2, pp. 125–133, Mar. 2017. <https://doi.org/10.1007/s40999-016-0090-6>
- [7] A. Hanna, *Concrete Formwork Systems*, 1st ed. Boca Raton: CRC Press, 1998. <https://doi.org/10.1201/9780203909690>
- [8] L. M. Vélez, “Permeabilidad y porosidad en concreto,” *TecnoLógicas*, no. 25, pp. 169–187, Dec. 2010. <https://doi.org/10.22430/22565337.131>

- [9] J. F. Arango-Londoño, “Diseño de pavimentos reforzados: evaluación de factores de carga y subresistencia por el método LRFD,” *TecnoLógicas*, no. 14, pp. 89-103, Jul. 2005.
<https://doi.org/10.22430/22565337.537>
- [10] J. F. Arango-Londoño, “Adoquines en concreto: propiedades físico-mecánicas y sus correlaciones,” *TecnoLógicas*, no. 16, p. 121-137, Jun. 2006.
<https://doi.org/10.22430/22565337.524>
- [11] M. Kapp y G. Girmscheid, “Empirical study reveals deficits in the choice of formwork,” en *Construction in the XXI century*, vol. 39. Zurich: *Edizioni Scientifiche Italiane*, 2006, pp. 172–173.
<https://doi.org/10.3929/ethz-a-005859139>
- [12] A. M. Jarkas, “The impacts of buildability factors on formwork labour productivity of columns,” *J. Civ. Eng. Manag.*, vol. 16, no. 4, pp. 471–483, Feb. 2010.
<https://doi.org/10.3846/icem.2010.53>
- [13] S. U. Dikmen y M. Sonmez, “An artificial neural networks model for the estimation of formwork labour” *J. Civ. Eng. Manag.*, vol. 17, no. 3, pp. 340–347, Sep. 2011.
<https://doi.org/10.3846/13923730.2011.594154>
- [14] R. H. Bordner et al., “Guide to Formwork for Concrete,” ACI Committee 347, United States, Reporte técnico, ACI 347R-14, Jul. 2001. Disponible en: [URL](#)
- [15] D. Lee, H. Lim, T. Kim, H. Cho, y K. I. Kang, “A Formwork Layout Model Based on Genetic Algorithm,” en *The 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC 2014)*, Sidney, 2014, pp. 474–479.
<https://doi.org/10.22260/ISARC2014/0063>
- [16] H. Al-Tabtabai, A. P. Alex, y R. James, “Slab formwork design using genetic algorithm,” *Durab. Build. Mater. Components*, pp. 2407–2418, 1999. Disponible en: [URL](#)
- [17] R. Ghowiba, “Horizontal Formwork Design Optimization and Selection System Using Genetic Algorithms”. (Tesis maestría). School of Science and Engineering. The American University in Cairo, Cairo, 2016. Disponible en: [URL](#)
- [18] T. Benoist, “Towards optimal formwork pairing on construction sites,” *RAIRO-Operations Res.*, vol. 41, no. 4, pp. 381–398, Oct. 2007.
<https://doi.org/10.1051/ro:2007035>
- [19] T. Benoist, A. Jeanjean, y P. Molin, “Minimum Formwork Stock problem on residential buildings construction sites,” *4OR - A Q. J. Oper. Res.*, vol. 7, pp. 275–288, Oct. 2009.
<https://doi.org/10.1007/s10288-008-0092-6>
- [20] Biruk, S. “Minimizing wall formwork cost in residential building construction”. *International Journal of Arts & Sciences*, vol. 6, no 3, pp. 355-362, 2013. Disponible en: [URL](#)
- [21] MiniZinc (2014) Monash University and Data61 Csiro. Disponible en: [URL](#)
- [22] Gecode (2016) Team. Gecode: Generic constraint development environment, 2016. Disponible en: [URL](#)
- [23] Coin-or/Cbc: Versión 2.9.9 (2018). <https://doi.org/10.5281/zenodo.1317566>
- [24] The Haifa Solvers Suite (2016). Technion Israel Institute of Technology. Disponible en: [URL](#)
- [25] FORSA. Catálogo técnico FORSA Acero. FORSA S.A. (2013). Disponible en: [URL](#)