Inteligencia Artificial

Estado del Arte: 2D Strip Packing Problem

Florencia Ramírez Sancristoful

27 de septiembre de 2023

Evaluación

Resumen (5 %):	____________
Introducción (5 %):	____________
Definición del Problema (10 %):	____________
Estado del Arte (35 %):	____________
Modelo Matemático (20 %):	____________
Conclusiones (20 %):	____________
Bibliografía (5 %):	____________
Nota Final (100 %):	____________

Resumen

El *2D Strip Packing Problem* es un problema de optimización en el cual un conjunto de elementos deben ser posicionados en una región definida cumpliendo ciertas condiciones con el objetivo de minimizar el espacio necesario para ajustar estos elementos. Se han propuesto una gran variedad de métodos y modelos para la resolución de este problema y variantes existentes, se presentan los métodos más relevantes y conocidos para luego realizar un análisis comparativo entre ellos con un enfoque la variante con rotaciones y sin cortes en guillotina.

1. Introducción

El *2D Strip Packing Problem* (2DPP) es un problema de gran interés estudiado por diversas industrias, derivado de los problemas de empaquetamiento y está enfocado en la optimización del espacio. Este problema consiste en ordenar, sin solapar, un conjunto de elementos rectangulares, cada uno con dimensiones conocidas, en una región de ancho fijo pero alto infinito, teniendo como objetivo final minimizar el área inutilizada de dicha región.

Este trabajo tiene por objetivo enfatizar la importancia del 2DSPP, presentando el estado del arte de este problema, dando a conocer las técnicas de resolución más relevantes existentes en la literatura actual. Analizando las ventajas y desventajas que tienen, para compararlas entre sí, midiendo la eficiencia y calidad de las soluciones encontradas por cada uno de estos métodos mencionados.

Dado que el 2DSPP tiene una gran cantidad de aplicaciones prácticas reales en la actualidad [6] [12], el desarrollo de este trabajo está impulsado por el deseo de contribuir a futuras investigaciones en la búsqueda e innovación de nuevos métodos de resolución, de modo que puedan adaptarse a los cambios del futuro y sigan brindando soluciones eficientes a este problema.

El resto de las secciones de este documento están estructuradas de la siguiente manera, en la sección 2 se explican los conceptos fundamentales del 2DSPP. En la sección 3 se presenta el Estado del Arte del problema. En la sección 4 se analizan algunos modelos matemáticos que entregan una solución al problema. Finalmente, en la sección 5 se presentan las conclusiones.

2. Definición del Problema

El *2D Strip Packing Problem* es un problema combinatorial de optimización derivado de los problemas de empaquetamiento [10], en el cual se deben ajustar una serie de elementos rectangulares, con un ancho w y alto h definido, en una región rectangular de ancho W y alto infinito, minimizando el área total inutilizada de la región o minimizando la altura total de la región. Este es un problema NP-Complejo conocido con bastantes aplicaciones en diversas industrias, incluyendo industrias madereras, vidrieras, metalúrgicas, textiles, entre otras [8] [12].

Al ser un problema NP-Complejo, ha sido estudiado de manera amplia con anterioridad y una de las limitaciones, o dificultades, que ha presentado es que los algoritmos de resolución propuestos en la literatura se dividen en tres tipos, los métodos exactos, los métodos heurísticos y los métodos híbridos [3] [5] [10] [14].

Los métodos exactos resuelven este problema encontrando la solución óptima, pero no pueden resolver instancias de mayor tamaño eficientemente. Los métodos heurésticos buscan una solución de buena calidad, pero no aseguran que la solución sea óptima. Los métodos hÃbridos combinan diversos métodos para encontrar una solución de buena calidad. Por lo que, hasta el momento no se han encontrado formas de obtener la solución óptima y resolver este problema de manera eficiente en las situaciones reales más complejas [11].

2.1. Variables, restricciones y objetivos

Antes de definir las variables de este problema, es necesario identificar las constantes ya existentes. Las cuales corresponden a la cantidad de elementos rectangulares a posicionar, N, el ancho definido de la región, W, y las dimensiones de cada elemento i, $w_i * h_i$.

Una vez identificadas estas constantes, se pueden definir tres variables para la resolucón de este problema. Dos variables correspondientes a la posición que tomará el elemento i, x_i e y_i, y una variable correspondiente al indicador de rotación del elemento i, r_i. Además, estas variables se ven restringidas por las siguientes condiciones:

1. Cada elemento debe estar posicionado dentro de los márgenes de la región.
2. Los elementos no pueden solaparse, en otras palabras, dos elementos no pueden estar en la misma posición.

El objetivo de este problema es acomodar todos estos elementos minimizando la cantidad de espacio inutilizado en la región, esto puede ser logrado minimizando la altura de la región.

2.2. Problemas relacionados y variantes más conocidas

El *2D Strip Packing Problem* es considerado un subproblema de los llamados *Packing Problems* (PP) [1] [8], también conocidos como problemas de empaquetamiento en español, en los cuales se deben acomodar objetos en contenedores específicos y tiene dos posibles objetivos, utilizar la menor cantidad de contenedores posibles o ajustar todos estos objetos en un único contenedor desperdiciando la menor cantidad de espacio posible.

Algunas de las características que pueden cambiar en las variantes de este problema son las dimensiones, tipos de cortes o patrones, conocimiento previo de los elementos, formas, ortogonalidad y rotación [6] [10].

Entre estas posibles variantes, son normalmente conocidos cuatro subproblemas que consideran limitaciones relacionadas a la rotación y corte de los elementos [6] [8]. En relación a la rotación de estos elementos, tienen una orientación fija que no se puede modificar (O) o pueden ser rotados en 90°(R), y en relación al tipo de corte de estos, se deben acomodar para permitir cortes en guillotina (G) o pueden ser posicionados libremente (F). Considerando estas características, se identifica la variante RF que tiene la menor cantidad de restricciones y, en consecuencia, es considerada como la más fácil de resolver. Si se fija la orientación de estos elementos, no pueden ser rotados, se define la variante OF. Finalmente, si se imponen los cortes en guillotina se pueden identificar las versiones RG y OG, permitiendo la rotación de los elementos o fijando su orientación, respectivamente.

También, se distinguen dos categorías respecto al conocimiento previo sobre los elementos [10]. La primera categoría es conocida como *online* y en esta no se tiene conocimiento sobre cómo son los siguientes elementos a posicionar, implica que una vez posicionado un elemento este ya no puede ser movido. En cambio, la segunda categoría, *offline*, es la más común en investigaciones y se distingue por tener conocimiento previo de todos los elementos a posicionar, lo cual hace posible definir una secuencia sobre cómo serán colocados.

Dos de los problemas más similares son el *Rectangular Packing Problem* y el *Bin Packing Problem*. En el primero se deben ordenar una serie de rectángulos en una región de ancho y largo fijo con el objetivo de maximizar el área de estos rectángulos ajustados [2]. Por otro lado, en el segundo se debe ajustar una serie de elementos en una cantidad finita de contenedores de ancho y alto fijo, tiene como objetivo minimizar la cantidad de contenedores utilizados [8].

3. Estado del Arte

El *2D Strip Packing Problem* (2DSPP) es uno de los problemas de empaquetamiento más simples que surge de la necesidad de reducir los costos operacionales de las industrias transformadoras de materias primas [6], por lo cual ha sido estudiado intensamente y se han propuesto diversos métodos para su resolución, los cuales pueden ser divididos en tres categorías, métodos exactos, heurísticos e híbridos.

Los métodos exactos se caracterizan por encontrar la solución óptima, los cuales funcionan cuando se tienen instancias más pequeñas a resolver ya que cuando intentan resolver instancias más grandes los tiempos de ejecución son muy altos, como estos tiempos de ejecución son tan altos, no se considera como un método eficiente. Los métodos heurísticos no aseguran que la solución encontrada sea óptima, tampoco reconocen la solución óptima, pero se caracterizan por entregar una solución de buena calidad con tiempos de ejecución considerados bajos y, a diferencia de los métodos exactos, los hace capaces de encontrar una solución cuando se tienen instancias de mayor tamaño. Por último, los métodos híbridos combinan aspectos de estos otros métodos para entregar soluciones factibles a este problema.

La mayoría de las investigaciones recientes se enfocan en los métodos heurísticos e híbridos, debido a que una de las principales limitaciones existentes es que no se ha logrado proponer un algoritmo de resolución que entregue la solución óptima a este problema para instancias de mayor tamaño. Las instancias de menor tamaño son resueltas de manera eficiente por los métodos ya existentes, por lo que no se considera necesario buscar algoritmos más eficientes y se enfocan principalmente en buscar nuevas técnicas para la resolución de instancias más grandes, que no pueden ser resueltas por los métodos exactos.

3.1. Métodos exactos

Los métodos exactos garantizan que la solución entregada es la óptima y normalmente están basados en modelos matemáticos [10]. En comparación a todos las investigaciones hechas sobre los otros dos métodos, existen menos investigaciones sobre los métodos exactos y, debido a las restricciones computacionales de tiempos de ejecución y memoria, no son capaces de resolver instancias de gran tamaño eficientemente [3], lo que contribuye a la cantidad limitada de información reciente y actualizada sobre estos métodos.

La mayoría de estos métodos se basan en la estrategia *Branch and Bound*, una forma de *backtracking* que genera un árbol con todas las posibles soluciones y corta las ramas que dejan de ser óptimas. Basándose en una adaptación para el *Contiguous Bin Packing Problem* (CBP) y modelos matemáticos propuestos [9], surgen los algoritmos *STAIRCASE* y *G-STAIRCASE* [7] que utilizan esta metodología.

El primero obtiene un límite inferior correspondiente a una altura H propuesta, luego agrega elementos con dimensiones $1 * 1$ hasta obtener una instancia de *Perfect Packing* (PP), casos perfectos en los que no existen áreas desperdiciadas, luego revisa si la instancia generada puede ser resuelta con una estrategia de posicionamiento en escalera considerando instancias PP. Si encuentra una solución válida, termina la ejecución y H corresponde a la solución óptima, en caso contrario, incrementa en valor de H en uno y repite este proceso. El segundo algoritmo también obtiene un límite inferior correspondiente a la altura H, pero no agrega elementos extra, luego revisa si puede ser resuelto utilizando la misma estrategia, este caso permite los espacios vacíos para generar en la región bajo la escalera y, al igual que el anterior, si encuentra una solución válida termina, sino aumenta H en uno.

Ambos algoritmos logran resolver la mayoría de las instancias de prueba con un máximo de 200 recángulos en menos de una hora, y concluyen que estos algoritmos son una alternativa a los propuestos anteriormente, considerando que la mayoría de los algoritmos existentes hasta ese momentos estaban enfocados en la variante OF.

3.2. Métodos heurísticos

Estos métodos son una alternativa a los métodos exactos, dado que los últimos no son capaces de entregar una solución de manera eficiente, los métodos heurísticos, también conocidos como métodos aproximados, son capaces de brindar soluciones de buena calidad en bajos tiempos. A diferencia de los métodos exactos, no buscan la solución óptima y, si la encuentran, no son capaces de reconocerla.

La primera heurística propuesta en la literatura fue *Bottom Left* (BL) [10] y es una de las estrategias más reconocidas, pertenece al conjunto de estrategias que selecciona un elemento y después identifica el espacio disponible más apto para posicionarlo. El objetivo de BL es posicionar cada elemento lo más abajo y a la izquierda posible y tiene la ventaja de ser considerado eficiente y simple, pero una de sus mayores desventajas es su incapacidad de rellenar los espacios vacíos rodeados de elementos ya posicionados. En base a esta metodología surgen *Bottom Left Fill* (BLF), una variante que considera estos espacios que eran desperdiciados, e *Improved Bottom Left* (iBL), una variante mejorada de BL que le da prioridad a los movimientos hacia abajo y los mueve a la izquierda lo suficiente para que sigan bajando.

También existen las estrategias que están enfocadas en asignar el mejor elemento a un espacio libre determinado y su primera heurística propuesta fue *Best Fit* (BF) [13]. Esta selecciona un espacio libre lo más abajo posible y luego busca un elemento que quepa perfectamente, si no existe este elemento se escoge el elemento más grande que quepa en dicho espacio y tiene tres posibles posiciones, lo más a la izquierda posible, junto al elemento de mayor altura o junto al elemento de menor altura. Con el método BF surgió el concepto de *skyline*, por lo que se proponen métodos mejorados que pueden resolver eficientemente instancias de tamaño grande, *Three-way Best Fit* (TwBF) que agrega tres nuevos criterios para posicionar los elementos, y *Fast Layer-*

Based Heuristic (FH) que propone mantener la línea del *skyline* lo más plana posible utilizando una llamada *fitness function*. Esta función utiliza una escala de 4 valores para determinar qué elemento es el más apto para asignarlo a un espacio libre específico, siendo 0 el peor y 4 el mejor. Utilizando esta *fitness function* y la idea de mantener el *skyline* lo más plano posible se proponen diversas mejoras como *Intelligent Search Algorithm* (ISA) y *Hybrid Demon Algorithm* (HDA), ambos usando una escala con cinco posibles valores, *Simple Randomized Algorithm* (SRA) y *Efficient Intelligent Search Algorithm* (IA) [16], los cuales ocupan una escala de ocho valores.

Otros métodos que no entran en estas dos categorías son *Heuristic Recursive Algorithm* (HR) y *Deterministic Heuristic Algorithm* (DHA) [3]. El primero utiliza la técnica dividir y conquistar y en cada iteración divide el espacio vacío en dos subespacios, soluciona recursivamente cada uno y finalmente combina las soluciones, su principal desventaja es el orden en que se posicionan los elementos afecta la eficiencia de este algoritmo. El segundo se compone de dos fases, la primera fase utiliza una estrategia *greedy* para posicionar los elementos y la segunda fase aplica una búsqueda parcial utilizando árboles para encontrar la posición que minimice la altura.

3.3. Métodos híbridos

Los métodos híbridos, como su nombre lo indica, brindan soluciones de buena calidad de manera eficiente combinando el uso de diversos algoritmos. Surgen bajo la idea de obtener una solución inicial que luego va a ser mejorada usando otro algoritmo. Es mÃ¡s común el uso de *Genetic Algorithm* (GA), *Simulated Annealing* (SA) o *Tabu Search* (TS) para generar la solución inicial y son normalmente usados con los algoritmos BL, BF y todas sus versiones mejoradas.

GA utiliza una analogía de la selección natural, a todas las posibles soluciones de un problema se le asigna un valor basándose en una *fitness function* [11], lo que finalmente se utiliza para generar una secuencia de los elementos a posicionar. SA surge como una variante del método *Hill-Climbing* (HC) [4], una técnica de búsqueda local que se mueve entre las soluciones de un vecindario, si se encuentra una mejor solución realiza el cambio y sigue con la búsqueda, para que no estancarse en un mínimo local, en vez de solo escoger soluciones mejores que la actual, acepta soluciones peores. TS, al igual que SA, es una técnica de búsqueda local derivada de HC [15], a la cual se le adjunta una memoria a corto plazo, para guardar las últimas soluciones y evitar los ciclos entre las mismas.

3.4. Comparaciones entre métodos

Los resultados computacionales de los métodos exactos *STAIRCASE* y *G-STAIRCASE* [7] en el cuadro 1 muestran que ambos algoritmos pueden resolver instancias de hasta 200 elementos, y la solución óptima entregada es igual, o muy cercana, al límite inferior calculado, pero entre ambos algoritmos *STAIRCASE* obtiene mejores resultados agregando los elementos indicados por la columna $1 * 1$, destancando los menores tiempos de ejecución.

Instancia	N	W	LW	H	$1*1$	*STAIRCASE*	*G-STAIRCASE*
beng01	20	25	30	30	9	**0.08**	**0.08**
beng02	40	25	57	57	5	0.12	**0.10**
beng03	60	25	84	84	10	**0.10**	**0.10**
beng04	80	25	107	107	2	**0.08**	0.13
beng05	100	25	134	134	20	**0.08**	0.13
beng06	40	40	36	36	20	**0.10**	0.12
beng07	80	40	67	67	7	**0.11**	**0.11**
beng08	120	40	101	101	13	**0.17**	0.18
beng09	160	40	126	126	32	**0.23**	0.41
beng10	200	40	156	156	23	**0.81**	3.53

Cuadro 1: Tiempo de ejecución [s] de los algoritmos para entregar la solución óptima.

Las comparaciones [4] entre los métodos híbridos y heurísticos destacan los métodos que obtuvieron la menor diferencia relativa entre la solución entregada y la solución óptima. Considerando todas las instancias presentadas en el cuadro 2 y sus resultados respectivos se concluye que el método híbrido SA + BLF obtuvo el mejor rendimiento, mientras que el método heurístico BL obtuvo las mayores diferencias. Estos resultados son acorde a lo presentado en la literatura, el algoritmo BL fue uno de los primeros propuestos, BLF es una versión mejorada de este y mejora la solución entregada con SA.

Método	C1	C2	C3	C4	C5	C6	C7
GA + BL	6	10	8	9	11	15	21
SA + BL	**4**	7	7	6	6	7	13
HC + BL	9	18	11	14	14	20	25
BL	17	31	24	18	18	21	29
GA + BLF	**4**	7	**5**	**3**	4	4	5
SA + BLF	**4**	**6**	**5**	**3**	**3**	**3**	**4**
HC + BLF	7	10	7	7	6	7	7
BLF	11	16	12	5	5	5	5

Cuadro 2: Distancia relativa (%) entre la solución entregada y la solución óptima.

Al comparar el algoritmo DHA [3] con otros métodos híbridos y heurísticos previos se destaca en el cuadro 3 las soluciones óptimas encontradas y se puede observar cómo este algoritmo nuevo supera a los demás y en los casos donde no encuentra la solución óptima está muy cercana a ella, mientras que en las instancias de mayor tamaño y complejidad es capaz de obtener esta solución óptima en la mitad de estas.

Instancia	N	W	H_{opt}	BF + TS	BF + SA	BF + GA	FH	DHA
C11	16	20	20	**20**	**20**	**20**	**20**	**20**
C23	25	40	15	16	16	16	**15**	**15**
C33	28	60	30	31	31	31	32	**30**
C42	49	60	60	62	61	62	61	61
C51	73	60	90	92	91	92	91	**90**
C63	97	80	120	122	122	122	121	**120**
C71	196	120	240	245	244	245	241	241

Cuadro 3: Soluciones entregadas por los métodos BF+TS, BF+SA, BF+GA, FH y DHA.

4. Modelo Matemático

Los modelos presentados en la literatura [3] [4] [7] [10] para la resolución de las diversas variantes del 2DSPP tienen en común algunos de los aspectos básicos más relevantes de este problema, por lo cual, en base a estos modelos se combinan y propone el siguiente modelo matemático para la resolución de la variante RF, que permite rotaciones y posiciones libres.

4.1. Constantes

La definición de este problema define tres constantes, las cuales representan el conocimiento previo de este problema y aportan a su resolución. La primera constante corresponde al ancho fijo de la región y, debido a la definición de medida, este valor debe ser un número entero mayor que 0.

$W =$ ancho predefinido de la región

La segunda constante corresponde a la cantidad total de elementos rectangulares que se deben ajustar en dicha región de ancho W, este valor también debe ser un número entero mayor que 0, al igual que el ancho, por la definición de cantidad, y la última constante corresponde a las dimensiones de cada uno de estos N elementos rectangulares, estas dimensiones están representadas de la forma $ancho * alto$.

$N =$ cantidad de elementos rectangulares

$w_i * h_i =$ dimensiones del elemento $i, \ \forall i \in [1, N]$

4.2. Variables

Considerando que se representa la variante RF, permitiendo rotaciones y posiciones libres, se definen tres variables de decisión. Dos de estas corresponden a la posición de la esquina inferior izquierda de los elementos, tomando un plano cartesiano como referencia, estas posiciones serán de la forma $(x_i,\, y_i)$.

$(x_i, y_i) =$ posición de la esquina inferior izquierda del elemento $i, \ \forall i \in [1, N]$

La última variable de decisión corresponde una variable binaria representando el índice de rotación de los elementos, indicando con los valores 1 y 0, si se rota o no dicho elemento, respectivamente.

$$r_i = \begin{cases} 1, & \text{se rota el elemento } i, \ \forall i \in [1, N] \\ 0, & \text{caso contrario} \end{cases}$$

4.3. Objetivos

Su objetivo principal es minimizar el área A sin utilizar de la región, lo cual puede ser logrado minimizando la altura H de dicha región.

$$\text{mín}\, H = \text{mín}\{\max_{i \in [1,N]} \{y_i + h_i + r_i * (w_i - h_i)\}\}$$

4.4. Restricciones

Esta variante define algunas condiciones que deben cumplir estas variables de decisión, por lo que se identifican tres restricciones en total, dos de estas son parte de la definición del problema y una de estas resulta de las variables de decisión.

Restricción 1: Los elementos no pueden superar el ancho W de la región.

$$x_i + w_i + r_i * (h_i - w_i) \leq W,\ \forall i \in [1, N] \quad (1)$$

La restricción considera la posible rotación del elemento i, si el índice de rotación es 0, el elemento i utiliza el espacio entre los puntos x_i y $x_i + w_i$ en el eje x del plano cartesiano. En cambio, si el índice es 1, significando que el elemento i es rotado en 90°, utilizará el espacio entre los puntos x_i y $x_i + h_i$.

Restricción 2: Los elementos no pueden solaparse entre sí.

$$\begin{aligned} &x_i + w_i + r_i * (h_i - w_i) \leq x_j \vee x_j + w_j + r_j * (h_j - w_j) \leq x_i \vee \\ &y_i + h_i + r_i * (w_i - h_i) \leq y_j \vee y_j + h_j + r_j * (w_j - h_{ij}) \leq y_i,\ \forall i, j \in [1, N], i \neq j \end{aligned} \quad (2)$$

Esta restricción aplica el mismo principio que la restricción anterior. Si se tienen dos elementos i, j existen cuatro posibles casos. El elemento i puede estar a la izquierda, derecha, debajo o sobre el elemento j. Mientras se cumpla al menos uno de los posibles casos, debido al operador lógico $\vee$, significa que los elementos no se solapan.

Restricción 3: Naturaleza de las variables.

$$x_i, y_i \geq 0,\ x_i, y_i \in \mathbb{Z},\ r_i \in \{0, 1\},\ \forall i \in [1, N] \quad (3)$$

Usando un plano cartesiano como referencia, se considera que el ancho de la región va desde el punto (0, 0) al punto (W, 0) y si alguna de las variables de posición de los elementos tiene un valor negativo, significa que fue colocado fuera de los márgenes establecidos de la región. Por la definición de las constantes, las dimensiones de los elementos corresponden a un número entero, no considera decimales. Por último, las variables del índice de rotación de los objetos corresponde a una variable binaria, lo que significa que sus únicos valores posibles son 0 o 1.

5. Conclusiones

El *2D Strip Packing Problem* es un problema con diversas aplicaciones prácticas reales, por lo cual es importante buscar soluciones buenas y eficientes para instancias de mayor tamaño y complejidad, ya que estas son las que más se asemejan a sus aplicaciones reales. Existen diversas técnicas enfocadas en las distintas variantes que existen, las restricciones que debe cumplir cada uno de estos métodos depende de la variante de estudio, considerando algunas más sencillas que otras. Los métodos heurísticos e híbridos son los más favorables, dado que logran resolver las instancias más parecidas a la realidad, pero aún no pueden obtener la solución óptima para estas instancias más complejas. Las investigaciones recientes respecto al problema están enfocadas en estos métodos y uno de los posibles trabajos a realizar en un futuro es crear nuevos métodos híbridos con las nuevas técnicas heurísticas más competitivas hasta la fecha.

Referencias

[1] İsmail Babaoğlu. Solving 2d strip packing problem using fruit fly optimization algorithm. *Procedia computer science*, 111:52–57, 2017.

[2] Mao Chen and Wenqi Huang. A two-level search algorithm for 2d rectangular packing problem. *Computers & Industrial Engineering*, 53(1):123–136, 2007.

[3] Kun He, Yan Jin, and Wenqi Huang. Heuristics for two-dimensional strip packing problem with 90 rotations. *Expert systems with applications*, 40(14):5542–5550, 2013.

[4] EBCH Hopper and Brian CH Turton. An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *European Journal of Operational Research*, 128(1):34–57, 2001.

[5] Alvaro Neuenfeldt Júnior, Julio Siluk, Matheus Francescatto, Gabriel Stieler, and David Disconzi. A framework to select heuristics for the rectangular two-dimensional strip packing problem. *Expert Systems with Applications*, 213:119202, 2023.

[6] Alvaro Neuenfeldt Júnior, Elsa Silva, Matheus Francescatto, Carmen Brum Rosa, and Julio Siluk. The rectangular two-dimensional strip packing problem real-life practical constraints: A bibliometric overview. *Computers & Operations Research*, 137:105521, 2022.

[7] Mitsutoshi Kenmochi, Takashi Imamichi, Koji Nonobe, Mutsunori Yagiura, and Hiroshi Nagamochi. Exact algorithms for the two-dimensional strip packing problem with and without rotations. *European Journal of Operational Research*, 198(1):73–83, 2009.

[8] Andrea Lodi, Silvano Martello, and Michele Monaci. Two-dimensional packing problems: A survey. *European journal of operational research*, 141(2):241–252, 2002.

[9] Silvano Martello, Michele Monaci, and Daniele Vigo. An exact approach to the strip-packing problem. *INFORMS journal on Computing*, 15(3):310–319, 2003.

[10] José Fernando Oliveira, Alvaro Neuenfeldt Júnior, Elsa Silva, and Maria Antónia Carravilla. A survey on heuristics for the two-dimensional rectangular strip packing problem. *Pesquisa Operacional*, 36:197–226, 2016.

[11] Jaya Thomas and Narendra S Chaudhari. Hybrid approach for 2d strip packing problem using genetic algorithm. In *Advances in Computational Intelligence: 12th International Work-Conference on Artificial Neural Networks, IWANN 2013, Puerto de la Cruz, Tenerife, Spain, June 12-14, 2013, Proceedings, Part I 12*, pages 566–574. Springer, 2013.

[12] Igor Vasilyev, Anton V Ushakov, Dong Zhang, and Jie Ren. Generalized multiple strip packing problem: Formulations, applications, and solution algorithms. *Computers & Industrial Engineering*, 178:109096, 2023.

[13] Jannes Verstichel, Patrick De Causmaecker, and Greet Vanden Berghe. An improved best-fit heuristic for the orthogonal strip packing problem. *International Transactions in Operational Research*, 20(5):711–730, 2013.

[14] Lijun Wei, Qian Hu, Stephen CH Leung, and Ning Zhang. An improved skyline based heuristic for the 2d strip packing problem and its efficient implementation. *Computers & Operations Research*, 80:113–127, 2017.

[15] Lijun Wei, Wee-Chong Oon, Wenbin Zhu, and Andrew Lim. A skyline heuristic for the 2d rectangular packing and strip packing problems. *European Journal of Operational Research*, 215(2):337–346, 2011.

[16] Lijun Wei, Hu Qin, Brenda Cheang, and Xianhao Xu. An efficient intelligent search algorithm for the two-dimensional rectangular strip packing problem. *International Transactions in Operational Research*, 23(1-2):65–92, 2016.