Un algoritmo de generación de filas y columnas para el manufacturer's pallet loading problem

Autoría ditelliana: Marenco, Javier (*Universidad Torcuato Di Tella. Escuela de Negocios*)

Publicado en: Memorias de las JAIIO (ISSN 2451-7496)

Fecha: 29/08/2024

Un algoritmo de generación de filas y columnas para el *manufacturer's pallet loading problem*

Javier Marenco[1][0000−0003−2694−4758]

Escuela de Negocios, Universidad Torcuato Di Tella, Av. Figueroa Alcorta 7350
(1428) Ciudad de Buenos Aires, Argentina
javier.marenco@utdt.edu

Resumen En este trabajo consideramos el *manufacturer's pallet loading problem*, que consiste en ubicar cajas rectangulares en un contenedor rectangular, de modo tal que cada caja se ubique en posición vertical y los lados de cada caja estén paralelos a los lados del contenedor. Existen diversos enfoques tanto heurísticos como exactos para este problema. En este trabajo presentamos un algoritmo de generación de filas y columnas para la formulación canónica de programación lineal entera de este problema. Reportamos experimentos computacionales sobre instancias reales, que muestran que este algoritmo permite encontrar soluciones óptimas para instancias que hasta ahora estaban abiertas.

Keywords: pallet loading problem · programación entera · row and column generation

El *manufacturer's pallet loading problem* (MPLP) consiste en ubicar cajas rectangulares en un contenedor rectangular, de modo tal que las cajas estén ubicadas en posición vertical y con sus lados paralelos a los lados del contenedor. Formalmente, una instancia del MPLP está dada por el largo $L \in \mathbb{Z}_{>0}$, el ancho $W \in \mathbb{Z}_{>0}$ y la altura $H \in \mathbb{Z}_{>0}$ del contenedor, y por el largo $\ell \in \mathbb{Z}_{>0}$, el ancho $w \in \mathbb{Z}_{>0}$ y la altura $h \in \mathbb{Z}_{>0}$ de cada caja. Todas las cajas tienen las mismas dimensiones. Una solución está dada por un *packing* bidimensional de las bases de las cajas (es decir, sus lados de dimensiones $\ell \times w$) en la base del contenedor (es decir, su lado de dimensiones $L \times W$), que se repite $\lfloor H/h \rfloor$ veces en capas horizontales (ver la Figura 1).

El MPLP es un problema computacionalmente difícil, e incluso no se sabe si su versión de decisión pertenece a la clase NP, dado que la entrada de este problema está dada por solamente seis enteros pero la salida potencialmente consiste de la ubicación de todas las cajas. Existen diversos algoritmos tanto exactos como heurísticos para este problema. Los algoritmos basados en programación lineal entera están basados en la formulación de Beasley para el *two-dimensional non-guillotine cutting problem* [2]. Esta formulación se basa en la observación de que toda instancia admite una solución óptima en la cual la esquina superior izquierda de la base de cada caja está ubicada en $I \times J$, donde

$$I = \{i\ell + jw : i, j \in \mathbb{Z}_+\} \cap \{0, \ldots, L\},$$
$$J = \{i\ell + jw : i, j \in \mathbb{Z}_+\} \cap \{0, \ldots, W\}.$$

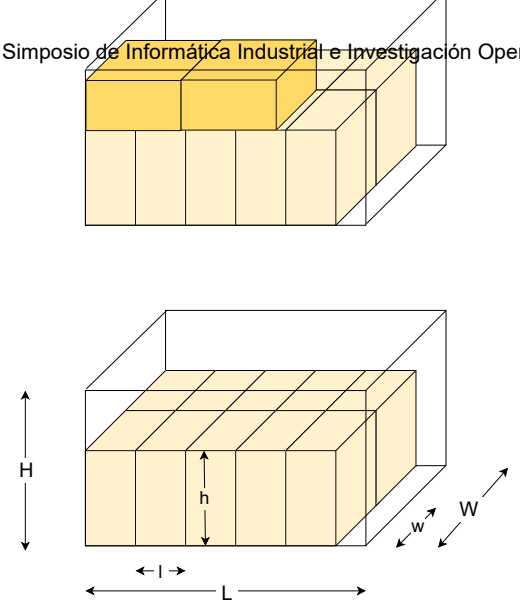

Figura 1. Datos de entrada del MPLP.

Decimos que una caja se ubica en *Configuración A* si su lado de longitud ℓ es paralelo al lado de longitud L del contenedor, y decimos que se ubica en *Configuración B* si su lado de longitud w es paralelo al lado de longitud L del contenedor. Para cada $(i,j) \in I \times J$ introducimos la variable binaria x_{ij} de modo tal que $x_{ij} = 1$ si se ubica una caja en Configuración A con la esquina superior izquierda de su base en el punto (i,j), e introducimos la variable binaria y_{ij} de modo tal que $y_{ij} = 1$ si se ubica una caja en Configuración B con la esquina superior izquierda de su base en el punto (i,j). Para cada $(i,j) \in I \times J$, definimos $C_A(i,j)$ como el conjunto de los puntos $(i',j') \in I \times J$ tal que si se ubica una caja en Configuración A en el punto (i',j') entonces esta caja toca al punto (i,j) de la base del contenedor. Definimos también $C_B(i,j)$ como el conjunto de los puntos $(i',j') \in I \times J$ tal que si se ubica una caja en Configuración B en el punto (i',j') entonces esta caja toca al punto (i,j) de la base del contenedor. Con estas definiciones, podemos considerar la siguiente formulación para el MPLP.

$$\text{máx} \sum_{i \in I} \sum_{j \in J} \lfloor H/h \rfloor (x_{ij} + y_{ij})$$

$$\sum_{(i',j') \in C_A(i,j)} x_{i'j'} + \sum_{(i',j') \in C_B(i,j)} y_{i'j'} \leq 1 \qquad \forall (i,j) \in I \times J$$

$$x_{ij}, y_{ij} \in \{0,1\} \quad \forall (i,j) \in I \times J$$

Esta formulación permite resolver instancias de tamaños pequeños y medianos del MPLP pero, dependiendo de la instancia, puede tener una cantidad demasiado grande de variables y restricciones. Se han presentado en trabajos previos un algoritmo de tipo branch and cut para esta formulación [1] utilizando el hecho de que esta formulación corresponde al modelo de conjunto estable de peso máximo en grafos, y un algoritmo basado en generación de columnas para resolver la relajación lineal [4] particionando la instancia en instancias más pequeñas del mismo problema.

Un problema fundamental que presenta la formulación anterior para plantear un algoritmo basado en generación de columnas es que un eventual problema maestro restringido que utilice un subconjunto de $I \times J$ no cuenta con todas las restricciones, dado que las restricciones también dependen de I y J. Más

aún, no resulta buena idea generar todas las restricciones asociadas con todos los elementos de $I \times J$, dado que este conjunto puede ser demasiado grande. Por estos motivos, en este trabajo exploramos un algoritmo de generación simultánea de filas y columnas para esta formulación, inspirado en las ideas presentadas en [3].

Presentamos las ideas principales de este algoritmo y describimos su implementación en Java utilizando Cplex para la resolución de los problemas de programación lineal involucrados en el algoritmo. Reportamos experimentos computacionales sobre instancias reales y de la literatura, que muestran que este enfoque es efectivo y permite resolver en forma óptima instancias que hasta ahora se encontraban abiertas.

Referencias

1. Alvarez-Valdes, R., Parreño, F., Tamarit, J.: A branch-and-cut algorithm for the pallet loading problem. Computers & Operations Research **32**(11), 3007–3029 (2005). https://doi.org/https://doi.org/10.1016/j.cor.2004.04.010
2. Beasley, J.E.: An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research **33**(1), 49–64 (1985). https://doi.org/10.1287/opre.33.1.49
3. Feillet, D., Gendreau, M., Medaglia, A.L., Walteros, J.L.: A note on branch-and-cut-and-price. Operations Research Letters **38**(5), 346–353 (2010). https://doi.org/https://doi.org/10.1016/j.orl.2010.06.002
4. Ribeiro, G.M., Lorena, L.A.N.: Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. Computers & Operations Research **34**(9), 2695–2708 (2007). https://doi.org/https://doi.org/10.1016/j.cor.2005.10.008