

## **PROBLEMA DE CARGA Y LOCALIZACIÓN DE CAJAS EN PALLETS CONSIDERANDO MÚLTIPLES PRODUCTOS: FORMULACIÓN Y MÉTODO DE SOLUCIÓN**

**Marcela C. González-Araya**

Departamento de Ingeniería Industrial, Facultad de Ingeniería, Universidad de Talca  
Camino a Los Niches km. 1, Curicó, Chile  
mgonzalez@utalca.cl

**René Ariel Bustos-Véliz**

Programa de Magíster en Gestión de Operaciones, Facultad de Ingeniería, Universidad de Talca  
Camino a Los Niches km. 1, Curicó, Chile  
mgonzalez@utalca.cl

**Luis G. Acosta-Espejo**

Departamento de Ingeniería Comercial, Universidad Técnica Federico Santa María  
Avenida Santa María 6.400, Vitacura, Santiago, Chile  
luis.acosta@usm.cl

### **RESUMEN**

En este trabajo se desarrolla y resuelve un modelo de programación lineal entera para el problema de carga y localización de cajas en pallets, considerando múltiples productos, de manera de minimizar los costos de operación del proceso de palletizado. El modelo formulado es NP-hard y fue resuelto a través de un método exacto y de una heurística basada en Algoritmo Goloso, usando 30 instancias. Los resultados obtenidos muestran que la heurística alcanza soluciones iguales o muy cercanas a las obtenidas con el método exacto, pero en un tiempo computacional mucho menor. En las instancias más complejas, el método exacto no obtuvo una solución factible dentro del tiempo computacional límite (3600 segundos). Por otro lado, la heurística siempre encontró una solución factible.

**PALABRAS CLAVE:** Carga de pallet, programación lineal entera, localización de pallets.

**Área principal:** AD & GP - IO en Administración y Gestión de la Producción; IND - IO en Industria; L & T - Logística y Transporte

### **ABSTRACT**

In this research, an integer linear programming model for box loading and pallets location problem, considering multiples products, is developed. The proposed model seeks to minimize operation costs from palletizing process. Because of the NP-hard nature of the model, it was solved using an exact method and a heuristic based on Greedy Algorithm for 30 instances. The obtained results show heuristic solutions are equal or very closed to the exact method solutions, however, these solutions were obtained in a lower computational time. For more complex instances, the exact method was not able to achieve a feasible solution in the maximum computation time (3600 seconds). In the other hand, the heuristic always found a feasible solution.

**KEYWORDS:** Pallet Load, Integer Linear Programming, Pallets Location.

AD & GP - OR in Administration & Production Management; IND - OR in Industry // IND - IO en Industria; L & T - Logistics and Transport

## 1. Introducción

Uno de los resultados de la estrategia de diversificación que adopta una empresa es el aumento de la variedad de productos con los que trabaja. Para facilitar el traslado y/o almacenamiento de estos productos se los empaican en cajas y, posteriormente, estas cajas se agrupan formando un pallet. El proceso de formar ordenadamente los pallets, respetando las características de los productos, se denomina “palletizado”.

En general, en el palletizado se realiza la conformación de cada pallet, esto es, se decide a cuál pallet asignar cada caja. Una vez que se completa un pallet, éste debe ser trasladado desde el área de palletizado hasta otra área ya sea de despacho o de almacenamiento, evitando el movimiento innecesario de algún pallet.

El problema de la formación o armado de pallets se conoce como *Pallet Loading Problem*, teniendo uno de sus primeros aportes con Hodgson (1982), quien planteó dos problemas: en el primero, llamado *The Manufacturer's Pallet Packing Problem*, tanto las cajas como los pallets son idénticos. En el segundo, llamado *The Distributor's Pallet Packing Problem*, las cajas tienen diferentes tamaños y el pallet es de tamaño estándar, en este problema se busca maximizar el volumen de cada pallet y minimizar el número de pallets utilizados.

El *Pallet Loading Problem* ha sido estudiado en su versión 2D (2 dimensiones) y 3D (3 dimensiones). En una aproximación preliminar, Gilmore y Gomory (1965) hicieron los primeros avances en la maximización de la utilización de una cierta área, siendo la base para el problema de carga de pallet en 2D. Posteriormente, se extendió a su forma en 3D (Tsai et al., 1993), considerando el volumen total del pallet.

En este trabajo se aborda el problema de decidir simultáneamente dónde realizar la palletización y cuál será la conformación del pallet, considerando múltiples productos. Para resolver el problema se propone un método de solución basado en Algoritmo Goloso.

Este artículo se organiza de la siguiente manera. En la Sección 2 se realiza una revisión de literatura. En la Sección 3 se detalla la problemática que motiva esta investigación. En la Sección 4 se describe el modelo matemático desarrollado y la heurística implementada para resolverlo. En la Sección 5 se muestran los resultados de la experimentación computacional realizada. Finalmente, en la Sección 6, se presentan las conclusiones y propuestas para futuras investigaciones.

## 2. Revisión de literatura

Problemas relacionados al palletizado se han estudiado en 2D y 3D. Uno de los primeros estudios relacionados a este tema es el de Gilmore y Gomory (1965). Estos autores trabajaron en el problema de corte en 2D. Este trabajo es una primera aproximación a los problemas de carga de pallet, ya que trata la optimización de un área específica cubierta por las cajas asignadas a un pallet, que es la base de los problemas de carga de pallet.

Hodgson (1982) mostró que el problema de la carga de pallets está relacionado con los problemas de corte y *Bin Packing Problem*. Este autor trabajó con el problema de carga de pallets en 2D, buscando maximizar el área cubierta por las cajas cargadas en el pallet. El problema se abordó mediante una combinación de programación dinámica y heurística. Dowsland (1987) planteó que los métodos de solución, hasta esa fecha, tienden a ser heurísticos, por lo que desarrolló un algoritmo de búsqueda exacta en árbol basado en un modelo de grafos.

Tsai et al. (1988) plantearon un modelo de programación lineal para determinar los patrones de apilamiento óptimos de las cajas en un pallet, abordando el problema en 2D. El modelo propuesto por estos autores permite variar el tamaño del pallet final, sin restringir el número de tipos de cajas que pueden ser cargadas. Tsai et al. (1988) no propusieron un método de solución específico para el modelo desarrollado. Chen et al. (1991) presentaron un modelo matemático de programación lineal entera mixta para *The Distributor's Pallet Packing Problem*, en donde se palletizan un conjunto de cajas de diferentes tamaños, con el fin de obtener la orientación de las cajas dentro del pallet, demostrando el funcionamiento del modelo a través de un ejemplo ilustrativo.

Un resumen de los artículos publicados hasta el año 1990 del *Pallet Loading Problem* fue presentado por Ram (1992), quién describe los modelos y procedimientos de solución utilizados.

Una de los trabajos para el palletizado en 3D fue expuesto por Tsai et al. (1993). Estos autores presentaron un modelo de programación entera mixta para el problema de carga de pallets en 3D,

permitiendo que el pallet pueda ser conformado con diferentes tamaños de cajas. En el modelo desarrollado por estos autores, no es posible asegurar que se complete totalmente un pallet, lo que puede ocasionar que aparezcan algunos espacios vacíos entre las cajas. Tsai et al. (1993) utilizaron un método de solución basado en el algoritmo *Branch-and-Bound*. Siguiendo en esta línea de investigación, Arghavani y Abdou (1996) desarrollaron un modelo de programación lineal entera que permite la optimización volumétrica del pallet, considerando cajas rectangulares y cuadradas de diferentes tamaños, con el fin de maximizar el número de cajas de diferentes tamaños que puede ser cargada en un pallet. En el trabajo de Arghavani y Abdou (1996), se propusieron dos casos de estudios, los que fueron resueltos a través del software de optimización LINDO.

Un método para cargar cajas del mismo tamaño en un pallet fue presentado por Liu y Hsiao (1997). Estos autores desarrollaron un modelo de programación entera para maximizar el grado de estabilidad y utilización del pallet y lo resolvieron con el software LINDO. Terno et al. (2000) plantearon el problema para más de un tipo de pallet, con el fin de minimizar el número de pallets utilizados. Estos autores desarrollaron una heurística basada en *Branch-and-Bound*.

Lashkari et al. (2002) propusieron un modelo de programación lineal entera multiobjetivo para la palletización. En este modelo consideraron un conjunto de piezas, máquinas y pallets con capacidades y compatibilidades conocidas. Estos autores buscaron maximizar la compatibilidad entre pieza-pallet y minimizar el tiempo de procesamiento total. Este modelo fue resuelto con un método de solución exacta. Chao et al. (2005) presentaron un caso práctico, donde diseñaron dos tipos de pallets mixtos, con el fin de reducir el exceso de carga de trabajo en horas punta. Chao et al. (2005) desarrollaron un modelo de programación entera, el cual fue resuelto a través del método de solución exacta. Alvarez-Valdés et al. (2005) desarrollaron un algoritmo *Branch-and-Cut* para el problema de palletización. Tres años más tarde, Martins y Dell (2008) presentaron diversos tipos de métodos de solución (heurísticas y algoritmos exactos) para tratar el *Pallet Loading Problem*.

Junqueira et al. (2012) formularon un modelo de programación lineal entera mixta para el problema de la carga de cajas rectangulares en contenedores, camiones, vagones o pallets, considerando la estabilidad horizontal/vertical y la fuerza de soporte de la carga. Los autores resolvieron el modelo usando CPLEX.

Los trabajos mencionados muestran las diferentes maneras de conformar un pallet, pero no consideran la ubicación de las cajas en una determinada área. Por otra parte, detallan distintos métodos para cargar los pallets, dentro de los que se encuentran la agrupación de cajas en pallets mixtos (Chao et al., 2005). El pallet mixto está compuesto por diferentes tipos de productos, pudiendo tener alguna característica en común, como, por ejemplo, el tamaño, color o código de las cajas. El problema de asignación y almacenamiento por clase de producto, el cual considera explícitamente la conformación de pallets mixtos, también han recibido atención en la literatura. Hausman et al. (1976) trabajaron en la asignación óptima de almacenamiento, comparando el desempeño operativo de las tres reglas de asignación de almacenamiento: aleatoria, dedicada y por clase. En el almacenamiento por clase, el inventario se realiza en base a ciertos criterios (demanda, tipo de producto, tamaño), asignando a cada clase un bloque de ubicaciones de almacenamiento.

Larson et al. (1997) emplearon los principios de almacenamiento basado en clase para aumentar la utilización del espacio físico y disminuir el manejo de materiales. Estos autores trabajaron directamente con heurísticas para determinar la distribución del almacén (sin un modelo matemático). Lai et al. (2002) consideraron un problema de diseño de bobina de papel, donde necesitan ser colocadas en las células de un almacén de diferentes clases. El problema radica en cómo diseñar los tipos de bobina de papel en el espacio celular, de forma tal de reducir al mínimo el costo total de transporte. El modelo planteado por Lai et al. (2002) es *NP-Hard*. Por esta razón, los autores desarrollaron una heurística basada en *Simulated Annealing* para resolver los casos complejos. Petersen et al. (2004) evaluaron el desempeño de las políticas de almacenamiento basado en clases y volumen, enfatizando el efecto del número de clases de almacenamiento y la estrategia de división de clases. Le-Duc y De Koster (2005) trabajaron en el problema para determinar las zonas óptimas de almacenamiento de las clases en cada pasillo del almacén.

### **3. Descripción del problema**

El problema consiste en decidir simultáneamente dónde realizar la palletización y cuál será la

conformación del pallet, considerando múltiples productos, observando limitantes como el tiempo, el espacio físico y la ubicación de las salidas del área para los pallets completos. Para este proceso, se asume que el palletizado se realiza en un área rectangular o cuadrada dentro de una planta.

Un esquema del problema se puede observar en la Figura 1, donde se tiene un conjunto de cajas (cuadradas y/o rectangulares), de diferentes tamaños, las cuales se deben palletizar en distintos períodos de tiempo  $t$ , con  $t = 1, \dots, n$ . Además de las cajas que llegan en cada período de tiempo, existen cajas en inventario, las que se pueden utilizar para completar los pallets. El palletizado se realiza en una zona especial, donde se cargan las cajas que llegan en cada período más las que se puedan sacar desde el inventario disponible. Como se observa en la Figura 1, la zona de palletizado tiene dos salidas para los pallets completos, tanto por el sector derecho, como por el inferior. Este supuesto es debido a que los pallets se trasladan con transpaletas o grúas horquilla y este tipo de grúas no pueden alcanzar los pallets de sectores internos de la zona de palletizado. De esta forma, cada pallet que se complete debe tener una salida por alguno de estos dos sectores.

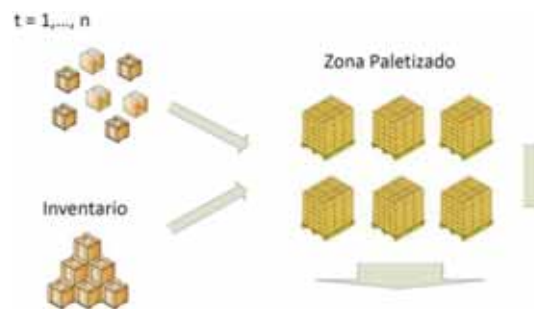


Figura 1. Esquema del problema

Un pallet que se encuentra en sectores internos del área de palletizado, tiene una salida sólo cuando no existen pallets localizados entre éste y el sector de salida. Esto se puede observar en la Figura 2, donde se representan cinco situaciones para ejemplificar la posible salida de un pallet.

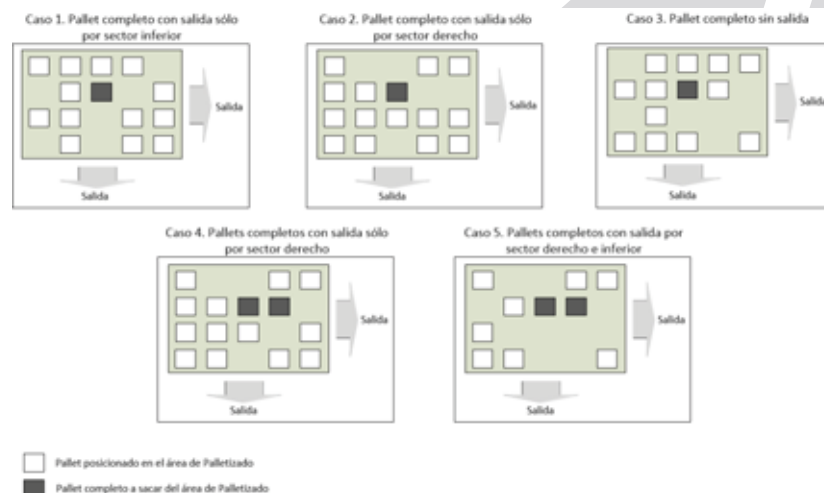


Figura 2. Situaciones de salida de un pallet

En el proceso de palletizado, generalmente se intentan crear pallets con un sólo tipo de producto, para así despachar o almacenar pallets homogéneos. Realizar esto en empresas que tienen un limitado espacio físico, puede provocar que se llene el área de palletizado con pallets antes del término de la jornada laboral, siendo muchos de éstos pallets incompletos. Si, además, se estuviese trabajando con múltiples productos, esta área se llena más rápido todavía, complicando aún más el proceso. Esta situación puede generar un cuello de botella si, para obtener más espacio libre dónde palletizar, se requiere reordenar las cajas en los pallets y conformar pallets mixtos.

Por otro lado, si se conforman solo pallets mixtos, para así minimizar el número de pallets, y por ende, minimizar el espacio utilizado, habrá dificultad para realizar los despachos. Esto, ya que será más difícil buscar los productos dentro de un pallet mixto en las áreas de almacenamiento.

Debido a las situaciones operacionales descritas anteriormente, la complejidad del problema de formar los pallets se incrementa a medida que aumenta el número de productos. Si, además, se debe decidir dónde realizar la carga de pallet (en qué pallet dentro del sector de palletizado dejar cada caja), el problema resulta aún más complejo.

En este trabajo se extiende el *Pallet Loading Problem* para determinar de manera simultánea dónde realizar el proceso de palletizado, siendo una contribución a los problemas existentes en la literatura. En este trabajo se integran conceptos tratados de manera individual en la literatura: pallets incompletos, presentado por Tsai et al. (1993), pallets mixtos, introducidos por Chao et al. (2005), carga de cajas del mismo tamaño y minimización pallets utilizados, trabajado por Liu y Hsiao (1997), almacenamiento basado en clases, presentado por Lai et al. (2002), Le-Duc y De Koster (2005), sectorización de posiciones por clase, presentado Hausman et al. (1976). Además de abarcar estos conceptos, se integran otros tales como la reutilización de la posición una vez que se completa el pallet y las condiciones de salida de un pallet completo.

#### 4. Formulación y método de solución para el problema de carga y localización de cajas en pallet con múltiples productos

Para abordar el problema propuesto se propone crear dos tipos de pallets, donde cada pallet contiene cajas de un mismo tamaño: se conforman pallets con un único tipo de producto al cual se llamará pallet de producto y pallets con diferentes tipos de productos, al que se llamará pallet mixto. De esta forma se reduce la complejidad del *Pallet Loading Problem*, teniendo sólo dos tipos de pallets que se pueden conformar. Con este abordaje, el problema se asemeja al *The Manufacturer's Pallet Packing Problem* planteado por Hodgson (1982).

Para llevar a cabo esta estrategia para la carga de pallet, se debe realizar un análisis previo, donde se debe determinar cuántas cajas conforman un cierto tipo pallet, sea éste un pallet de producto o uno mixto. Dado que los pallets mixtos están formados con cajas de igual tamaño, es conocida la cantidad de cajas que conforman un pallet mixto.

La conformación de pallets mixtos se justifica, ya que si en el último período de palletizado, existe una cantidad de cajas de un producto menor al máximo que conforma el pallet, en vez de dejar un pallet de producto incompleto con pocas cajas, se agrupa con otras cajas del mismo tamaño. De esta forma, se minimiza la cantidad de pallets a utilizar. Los pallets mixtos son una ventaja cuando la variedad de productos es elevada y se desea evitar tener una gran cantidad de pallets incompletos. La desventaja es que se tienen mayores tiempos para conformar las cargas y despachar los productos, ya que se van a tener que sacar cajas de diferentes pallets para agruparlos y cargarlos en los futuros despachos.

##### 4.1. Modelo matemático

Los supuestos para la formulación del modelo son los siguientes: i) El área de palletizado tiene forma rectangular. ii) Las cajas pueden ingresar al área de palletizado por cualquier lugar. iii) Los pallets completos deben salir del sistema sólo por la parte inferior o derecha del área de palletizado. iv) Las cajas utilizadas del inventario para completar un pallet, sólo deben ser colocadas en pallets de producto.

*Índices, conjuntos y subconjuntos*

$p, P$  Índice y conjunto para los productos a palletizar.

$t, i, T$  Índices y conjunto para los períodos en que ingresan los conjuntos de cajas al palletizado.

$a, j, A$  Índices y conjunto para las posiciones de ancho a utilizar en el área de palletizado.

$l, k, L$  Índices y conjunto para las posiciones de largo a utilizar en el área de palletizado.

$c, C$  Índice y conjunto correspondiente al tipo de cajas a cargar en el pallet.

$t_{max}$  Último período en el que ingresan conjuntos de cajas al sector de palletizado.

$Q = \{p \in P \mid DIF_p \geq 0\}$

$N_p = \{c \in C, a \in A, l \in L \mid DIF_p \geq 0 \wedge c = TIPO_p \wedge DISP_{cal} = 1\}$

$B_c = \{p \in P \mid TIPO_p = c\}$

$F = \{c \in C, a \in A, l \in L \mid DISP_{cal} = 1\}$

$D_i = \{i \in T \mid i > 1 \wedge i \leq t\}$

$H = \{i \in T \mid i < t_{max}\}$

### Parámetros

- $CAJ_{tp}$  Cantidad de cajas a palletizar del producto  $p$  en cada período  $t$ .  
 $SAL_p$  Cantidad de cajas de producto  $p$  que están disponibles en inventario.  
 $RP_p$  Cantidad de cajas que completan un pallet de producto  $p$ .  
 $RM_c$  Cantidad de cajas que completan un pallet mixto de tipo de caja  $c$ .  
 $TIPO_p$  Tipo de caja que utiliza el producto  $p$ .  
 $DIF_p$  Indicador proveniente de la suma de los  $CAJ_{tp}$  y  $SAL_p$  de cada producto  $p$  menos la cantidad de cajas que conforman un pallet ( $RP_p$ ).  
 $DISP_{cal} = 1$ , si hay disponibilidad en la posición  $a, l$  para ubicar un pallet del tipo  $c$ . 0, en c.  $c$ .  
 $M$  Número grande.  
 $CPOS_{al}$  Costo por utilizar la posición de ancho  $a$  y largo  $l$  para palletizar.  
 $PM$  Penalización monetaria por formar un pallet mixto.  
 $PI$  Penalización monetaria por pallet incompleto en el último período.  
 $PCM$  Penalización monetaria por la cantidad de productos a almacenar en un pallet mixto.  
 $BO$  Bonificación monetaria por completar un pallet de producto.

### Definición de variables

- $CJP_{tpal}$ : Cantidad de cajas a ubicar en los pallets de producto  $p$ , que se ordenan en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ .  
 $CJM_{tpal}$ : Cantidad de cajas a ubicar en los pallets mixtos de producto  $p$ , que se ordenan en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ .  
 $PENDP_{tpal}$ : Cantidad de cajas que faltan para completar un pallet de producto  $p$ , que se ordena en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ .  
 $PENDM_{tcal}$ : Cantidad de cajas del tipo  $c$  que faltan para completar un pallet mixto, que se ordena en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ .  
 $SALD_{tp}$ : Cantidad de cajas de producto  $p$  que se utilizan del inventario disponible en el período  $t$ .  
 $LIM_{tpal}$ : Tipos de producto  $p$  que se utilizan en los pallets mixtos, en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ .  
 $TIP_{al}$ : Cantidad de tipos de productos que se tienen dentro de un pallet mixto, en el último período y que se encuentran en la posición de ancho  $a$  y largo  $l$ .  
 $PUP_{tpal} = 1$ , si se utiliza el pallet de producto  $p$ , en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ . Valor 0 en caso contrario.  
 $PUM_{tcal} = 1$ , si se utiliza el pallet mixto de tipo de caja  $c$ , en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ . Valor 0 en caso contrario.  
 $PIP_{tpal} = 1$ , si está incompleto el pallet de producto  $p$ , en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ . Valor 0 en caso contrario.  
 $PIM_{tcal} = 1$ , si está incompleto el pallet mixto de tipo de caja  $c$ , en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ . 0, en caso contrario.  
 $CO_{tal} = 1$ , si hay un pallet, en la posición de ancho  $a$  y largo  $l$ , en el período  $t$  que está completo. Valor 0 en caso contrario.  
 $SDER_{tal} = 1$ , si el pallet, en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ , tiene salida del sistema por el sector derecho. Valor 0 en caso contrario.  
 $SABA_{tal} = 1$ , si el pallet, en la posición de ancho  $a$  y largo  $l$ , en el período  $t$ , tiene salida del sistema por el sector inferior. Valor 0 en caso contrario.

A continuación se presenta la formulación matemática del modelo.

$$\text{Minimize } z = \sum_{al} CPOS_{al} \times (\sum_{tp} PUP_{tpal} + \sum_{tc} PUM_{tcal}) + \sum_{tcal} PUM_{tcal} \times PM + \\
 PI \times (\sum_{pal} PIP_{t_{max}pal} + \sum_{ccl} PIM_{t_{max}cal}) + \sum_{al} TIP_{al} \times PCM - \\
 \sum_{tpal} (PUP_{tpal} - PIP_{tpal}) \times BO \quad (1)$$

Sujeto a:

$$\sum_{al} (CJP_{tpal} + CJM_{tpal}) - SALD_{tp} = CAJ_{tp}, \forall t, p \quad (2)$$

$$\sum_t SALD_{tp} \leq SAL_p, p \in Q \quad (3)$$

$$\sum_{al} CJM_{tpal} \leq CAJ_{tp}, \forall t, p \quad (4)$$

$$CJM_{tpal} \leq M \times LIM_{tpal}, \forall t, p, a, l \quad (5)$$

$$TIP_{al} = \sum_{tp} LIM_{tpal}, \forall a, l \quad (6)$$

$$\sum_{i \leq t} CJP_{ipal} + PENDP_{tpal} = RP_p \times (PUP_{tpal} + \sum_{i \in D_t} (PUP_{(i-1)pal} - PIP_{(i-1)pal})), \forall t, a, l, p \in N_p \quad (7)$$

$$\sum_{p, i \leq t} CJM_{ipal} + PENDM_{tcal} = RM_c \times (PUM_{tcal} + \sum_{i \in D_t} (PUM_{(i-1)cal} - PIM_{(i-1)cal})), \forall t, a, l, p \in B_c, c \in F \quad (8)$$

$$PIP_{tpal} \times M \geq PENDP_{tpal}, \forall t, a, l, p \in N_p \quad (9)$$

$$PIP_{tpal} \leq PENDP_{tpal}, \forall t, a, l, p \in N_p \quad (10)$$

$$PIM_{tcal} \times M \geq PENDM_{tcal}, \forall t, a, l, c \in F \quad (11)$$

$$PIM_{tcal} \leq PENDM_{tcal}, \forall t, a, l, c \in F \quad (12)$$

$$PUP_{tpal} \geq PIP_{tpal}, \forall t, a, l, p \in N_p \quad (13)$$

$$PUM_{tcal} \geq PIM_{tcal}, \forall t, a, l, c \in F \quad (14)$$

$$\sum_p PUP_{tpal} + \sum_c PUM_{tcal} \leq 1, \forall t, a, l \quad (15)$$

$$\sum_p PIP_{tpal} + \sum_c PIM_{tcal} \leq 1, \forall t, a, l \quad (16)$$

$$\sum_p (PUP_{tpal} - PIP_{tpal}) + \sum_c (PUM_{tcal} - PIM_{tcal}) = CO_{tal}, \forall t, a, l \quad (17)$$

$$SDER_{tal} + SABA_{tal} \geq CO_{tal}, \forall t, a, l \quad (18)$$

$$1 - CO_{tal} + \sum_{k>l} (\sum_p PIP_{tpak} + \sum_c PIM_{tcak}) \leq M \times (1 - SDER_{tal}), \forall a, l, t \in H \quad (19)$$

$$1 - CO_{tal} + \sum_{j>a} (\sum_p PIP_{tpjl} + \sum_c PIM_{tcjl}) \leq M \times (1 - SABA_{tal}), \forall a, l, t \in H \quad (20)$$

$$1 - CO_{tal} + \sum_{k>l} (\sum_p PIP_{tpak} + \sum_c PIM_{tcak}) \geq (1 - SDER_{tal}), \forall a, l, t \in H \quad (21)$$

$$1 - CO_{tal} + \sum_{j>a} (\sum_p PIP_{tpjl} + \sum_c PIM_{tcjl}) \geq (1 - SABA_{tal}), \forall a, l, t \in H \quad (22)$$

$$CJP_{tpal}, CJM_{tpal}, PENDP_{tpal}, PENDM_{tcal}, SALD_{tp}, LIM_{tpal}, TIP_{al} \in Z^+, \forall t, p, c, a, l \quad (23)$$

$$PUP_{tpal}, PUM_{tcal}, PIP_{tpal}, PIM_{tcal}, CO_{tal}, SDER_{tal}, SABA_{tal} \in \{0, 1\}, \forall t, p, c, a, l \quad (24)$$

La función objetivo (1) busca minimizar los costos en el proceso de palletizado. La primera expresión corresponde al costo asociado por utilizar una cierta posición en el área de palletizado, se considera que la posición que está más cerca de la salida tiene un menor costo. La segunda expresión es una penalización a la conformación de pallets mixtos. El tercer ítem corresponde a una penalización por los pallets incompletos del último período. Con el objetivo de homogeneizar la formación de pallets mixtos, se busca minimizar la cantidad de cajas, con diferentes productos, lo cual está representado en la cuarta expresión. Finalmente, en la última expresión de la función objetivo, se bonifica la formación de pallets completos de producto, para dar mayor importancia a la conformación de pallets de productos.

Las restricciones (2) establecen que la cantidad de cajas a almacenar en los pallets de productos y mixtos, de un cierto producto  $p$ , es igual a la cantidad de cajas que ingresa al área en cada período, más la cantidad de cajas provenientes del inventario. Las restricciones (3) garantizan que la cantidad de cajas a utilizar de cada producto proveniente del inventario debe ser menor o igual al número de cajas disponibles. Las restricciones (4) limitan la cantidad de cajas a almacenar en pallets mixtos, la cual debe ser menor o igual a lo que llega en cada período, esto permite utilizar las cajas del inventario sólo para pallets de productos. Las restricciones (5) permiten clasificar si un pallet es mixto, para así, en las restricciones (6), tener una cuenta del número de productos diferentes que conforman un pallet mixto en una determinada posición.

Las restricciones (7) establecen que la cantidad de cajas que se va a cargar en un pallet más las que faltan para completarlo, debe estar limitada a la capacidad máxima del pallet de producto. En esta misma restricción, se da la posibilidad de reutilizar esa posición al momento de completar un pallet. Las restricciones (8) son similares a las restricciones (7), pero para el caso de los pallets mixtos.

Las restricciones (9) - (12) reconocen pallets incompletos cuando hay cajas faltantes para completar un pallet, tanto en los pallets de producto, como en los mixtos. Con las restricciones (13) y (14) se reconoce un pallet incompleto de producto y mixto sólo si éste es utilizado. Las restricciones (15) aseguran que una posición pueda ser utilizada por un sólo pallet, ya sea de producto o mixto, en cada período. Las restricciones (16) cumplen la misma función que las

restricciones (15), pero para los pallets incompletos. Las restricciones (17) establecen que si un pallet, sea de producto o mixto, está siendo utilizado y no está incompleto, se clasifica como un pallet completo.

Es necesario considerar restricciones que indiquen por dónde deben salir los pallets completos del área de palletizado. En las restricciones (18) se tiene que un pallet completo debe tener como mínimo una salida, ya sea por el sector inferior o por el derecho. En las restricciones (19) - (22) se activan las salidas por cada sector cuando las posiciones contiguas respectivas estén vacías o completas en cada período. Finalmente, las restricciones (23) y (24) son las restricciones de integralidad y señalan la naturaleza de las variables de decisión.

Cabe hacer notar que, las restricciones (7) y (8) son similares a las presentadas en el modelo de Lai et al. (2002), las cuales son consideradas *NP-Hard* en su modelo. Además, trabajos como el de Hodgson (1982) y Tsai et al. (1993), hacen referencia a que el *Pallet Loading Problem* cae en la categoría de problema *NP-Hard*. Por esta razón, será necesario desarrollar una heurística, que permita obtener buenas soluciones en menor tiempo computacional que el método exacto.

#### 4.2. Heurística basada en algoritmo goloso

En este trabajo se implementó un algoritmo goloso, con el fin de tratar los distintos costos y beneficios de la función objetivo, asegurando la factibilidad de la solución encontrada.

El pseudocódigo implementado para la resolución del problema se muestra en la Tabla 1.

Tabla 1. Pseudocódigo del algoritmo goloso (AG)

<b>Output</b>	
<i>sol<sub>ga</sub></i> : Solución del Algoritmo Goloso y <i>FO</i> : Valor de la función objetivo	
<b>Método</b>	
1.	Lectura de datos de entrada del modelo y <i>TRF</i> : Total de restricciones del modelo
2.	Crear variable <i>x</i> , la cual va a tener valor 1 si la restricción de salida de pallet completo se cumple, valor 0 en caso contrario
3.	Inicializar $t = 0$
4.	Mientras ( $t < T$ )
5.	Inicializar contador de restricciones factibles $r = 0$
6.	Inicializar variable $x = 0$
7.	Crear solución para el período $t$
8.	Asignar a $r$ el número de restricciones factibles que se tienen con la solución creada
9.	Si $r < TRF$
10.	Mientras ( $r < TRF$ )
11.	Realizar cambios de posición $a, l$ entre pallet completo infactible y pallet incompleto factible de menor valor de posición
12.	Si $x = 1$ Confirmar cambio
13.	Si $x = 0$
14.	Si hay más pallets incompletos factibles por comparar
15.	Cancelar cambio y comparar con siguiente pallet incompleto factible de menor valor de posición
16.	Si no hay más variables factibles por comparar
17.	Cancelar cambio y pasar a comparar entre pallets completos
18.	Si $r = TRF$
19.	Si se puede cambiar posición de pallet utilizado por no utilizado para disminuir costo $CPOS_{at}$
20.	Realizar cambios de posición $a, l$ de variables si es que no se infringen restricciones
21.	$t = t + 1$
22.	Asignar a <i>FO</i> el costo de la solución
<b>Fin</b>	

El dato sobre el número total de restricciones que tiene el problema, permite evaluar la factibilidad de las soluciones obtenidas en cada período.

Antes de realizar el primer paso del algoritmo expuesto en la Tabla 1, se debe efectuar un análisis previo de los datos de entrada. Con el fin de disminuir la complejidad computacional de la heurística se realizan modificaciones en la información de entrada en los productos  $p$  que puedan completar a lo menos un pallet de producto a lo largo de los  $t$  períodos de palletizado. Estos productos, se pueden seleccionar sumando la cantidad total de cajas que ingresan al área de



palletizado en cada uno de los  $t$  períodos ( $CAJ_{tp}$ ) más las cajas disponibles en inventario ( $SAL_p$ ). Si al restar la cantidad de cajas que completan un pallet ( $RP_p$ ), se obtiene un valor superior o igual a cero, entonces se puede conformar a lo menos un pallet de producto. El parámetro  $DIF_p$  representa esta operación, teniendo la siguiente fórmula:  $DIF_p = \sum_t CAJ_{tp} + SAL_p - RP_p, \forall p$ .

Cada producto  $p$  que pueda conformar como mínimo un pallet de producto, se va a duplicar en la información de entrada. De esta forma se tendrá un producto P1 que se ajuste exactamente a la capacidad del pallet y siga teniendo el parámetro  $DIF_{p1} \geq 0$ . Además, se tendrá otro producto P2 que corresponde al remanente de las cajas que no alcanzan para completar el pallet, que tendrá un valor  $DIF_{p2} < 0$ . Con esto, los productos que tengan el parámetro  $DIF_p \geq 0$ , van a ser destinados sólo para crear pallets de productos y los que tengan  $DIF_p < 0$  se asignan sólo a pallets mixtos.

Este pre-procesamiento permite minimiza la dificultad del problema de carga de pallet, teniendo como dato de entrada los productos que van a conformar pallets de productos y/o mixtos. Si bien, con esto se limita el conjunto de posibles soluciones, pudiendo llegar a obtener una solución que no sea óptima, también se reduce la complejidad computacional, reduciendo el tiempo de solución.

En el paso 7 se crea la solución en cada período  $t$ . Para eso, se debe realizar un reordenamiento de la información dentro del algoritmo. Primero, se ordenan los productos  $p$ , según la cantidad de cajas que se palletizan en cada período (parámetro  $CAJ_{tp}$ ), de mayor a menor, según cada tipo de caja ( $TIPO_p$ ).

Los pallets de productos y mixtos se conforman en cada período de la siguiente manera:

- En el caso en que  $DIF_p \geq 0$ , se asigna el producto  $p$  como pallet de producto, en la posición disponible con menor costo  $CPOS_{al}$ . En caso que el período  $t$  sea mayor a 1, se verifica si es que hay pallets incompletos del producto  $p$ , para asignar el número de cajas al pallet en esa posición.
- En el caso en que  $DIF_p < 0$ , se asigna el producto  $p$  como pallet mixto. Si es que no hay cajas en el área de palletizado del  $TIPO_p$ , se agrega el producto  $p$  con mayor valor del parámetro  $CAJ_{tp}$  en la posición disponible con menor costo ( $CPOS_{al}$ ). En caso de que ya se tengan productos del  $TIPO_p$  en el área de palletizado, se busca si hay un producto  $p$  con la cantidad  $CAJ_{tp}$  suficiente para completar el pallet y se agrega a la posición existente. Si no hay un producto con la cantidad suficiente para completar el pallet, se agrega la cantidad menor más cercana a la faltante, con el fin de reducir las cajas faltantes para completar el pallet.

Con los pasos anteriores se crea una solución en cada período  $t$ . Para los casos en que  $t > 1$ , debido a la posición en que se tienen los pallets completos, puede ocurrir que la restricción de salida de pallet no se cumpla, es por esto que se verifica el cumplimiento de todas las restricciones en los pasos 8 y 9 del algoritmo. Si es que no se cumplen todas las restricciones, se realiza un cambio de posición  $a, l$  entre la variable en donde se presenta la infactibilidad y el pallet incompleto con posición factible de menor costo  $CPOS_{al}$ . Si con el cambio ya no hay restricciones infactibles, se confirma el cambio. Si aún quedan restricciones infactibles, se compara con el pallet incompleto con posición factible que tenga el siguiente valor de costo  $CPOS_{al}$  menor. Esto se realiza sucesivamente hasta que se logre tener una solución factible. Si al terminar de comparar con todas las variables factibles el problema continúa siendo infactible, se compara con los pallets completos con menor valor  $CPOS_{al}$  (de la misma manera a la explicada anteriormente).

Una vez que se tiene una solución con todas las restricciones factibles, en el paso 18 se comprueba si es que se puede cambiar la posición de un pallet utilizado por uno no utilizado, de tal forma de disminuir el costo  $CPOS_{al}$  y mantener las restricciones factibles. De esta forma, se generan todas las soluciones para cada período  $t$ .

## 5. Experimentación computacional

El modelo matemático ha sido programado en el software de optimización IBM ILOG CPLEX Optimization Studio, versión 12.5 y la heurística ha sido programada en el software Eclipse SDK, versión 4.2.0. Todas las instancias computacionales fueron corridas en un computador con procesador Intel(R) Core(TM) i7, con memoria RAM de 12 GB.

Para comparar las soluciones obtenidas entre el método exacto (CPLEX) y el algoritmo goloso (GA), se definieron cinco tipos de casos. Para cada caso, se corrieron instancias considerando un horizonte de planificación de 6 períodos, es decir,  $t = 1, \dots, 6$ , por lo tanto se

realizaron 30 experimentaciones. Estas instancias se basaron en los datos obtenidos de una empresa chilena localizada en la Región del Maule. Para las instancias corridas en CPLEX, se estableció un tiempo límite de 3600 segundos. Este tiempo computacional se considera razonable dado que se trata de tomar decisiones operativas. Los tipos de casos considerados se describen en la Tabla 2.

Tabla 2. Tipos de casos para la carga de pallets

Casos	Productos	Productos Modificados	Cajas por período	Tipos de cajas
1	28	30	350	4
2	35	37	550	5
3	50	53	650	6
4	75	80	850	8
5	100	108	1000	10

La cantidad de productos que se presentan en la columna Productos Modificados de la Tabla 2, son los que obtienen cuando se duplican los productos  $p$  que pueden conformar como mínimo un pallet de producto (conforme explicado en la sección anterior). Esta cantidad se mantuvo constante en cada caso, independiente del horizonte de planificación.

Los resultados obtenidos en la experimentación se muestran en la Tabla 3.

Tabla 3. Comparación de resultados obtenidos con CPLEX y AG

Casos	Períodos	CPLEX					Algoritmo Goloso		
		#RS	FO	GAP (%)	TAS (seg)	TF (seg)	FO	DIF <sub>H</sub> (%)	TF (seg)
1	1	3	163	0.00	1	6	163	0.00	1
2	1	6	149	0.00	1	683	164	9.15	1
3	1	8	240	0.00	4	2673	255	5.88	1
4	1	9	334	6.46	21	3600	349	4.30	1
5	1	5	446	2.51	4	3600	461	3.25	2
1	2	7	316	5.06	3	3600	321	1.56	1
2	2	7	252	25.88	60	3600	269	6.32	1
3	2	38	362	33.83	165	3600	392	7.65	1
4	2	73	679	29.29	3250	3600	709	4.23	1
5	2	56	955	30.59	3150	3600	940	-1.60	2
1	3	33	392	43.41	3310	3600	422	7.11	1
2	3	43	368	51.13	3390	3600	409	10.02	2
3	3	103	516	51.42	3312	3600	555	7.03	2
4	3	75	1026	42.11	3522	3600	1015	-1.08	2
5	3	-	-	-	-	3600	1483	-	3
1	4	120	444	67.44	3418	3600	515	13.79	1
2	4	104	541	66.55	3541	3600	558	3.05	1
3	4	238	712	61.11	3600	3600	742	4.04	1
4	4	265	1493	48.25	3600	3600	1456	-2.54	2
5	4	-	-	-	-	3600	2095	-	3
1	5	246	613	80.20	3580	3600	621	1.29	1
2	5	240	705	76.69	3600	3600	658	-7.14	1
3	5	263	972	71.51	3580	3600	900	-8.00	1
4	5	316	1874	54.09	3600	3600	1745	-7.39	2
5	5	-	-	-	-	3600	2572	-	5
1	6	147	641	87.19	3540	3600	667	3.90	1
2	6	272	920	83.23	3600	3600	745	-23.49	1
3	6	-	-	-	-	3600	1088	-	1
4	6	-	-	-	-	3600	2020	-	2
5	6	-	-	-	-	3600	3156	-	6

En la Tabla 3, la columna #RS corresponde al número de soluciones encontradas hasta el momento TF, donde TF es el tiempo computacional, en segundos, en que se detiene CPLEX; FO es el valor de la función objetivo; TAS es el tiempo computacional, medido en segundos, que demora CPLEX en encontrar la mejor solución. Para el caso de CPLEX, el GAP representa la brecha entre la solución relajada y la mejor solución encontrada. Para el algoritmo goloso,  $DIF_H$  representa la brecha entre la mejor solución encontrada con AG ( $FO_{AG}$ ) y la mejor solución encontrada en CPLEX ( $FO_{CPLEX}$ ), tal como se muestra en la siguiente fórmula:

$$DIF_H = (FO_{AG} - FO_{CPLEX})/FO_{AG}$$

En la Tabla 3 se aprecia, cuando se considera un sólo período, que CPLEX encontró la solución final en los primeros (TAS < 25 seg). Este tiempo computacional va aumentando a medida que se consideran más períodos. Cuando se considera un sólo período, CPLEX obtiene solución óptima en 3 casos. En los casos 4 y 5, para un período, la ejecución de CPLEX se detiene en el tiempo límite TF.

Es importante destacar que, a partir del período 3, para el caso 5 no se obtienen soluciones iniciales con CPLEX dentro del tiempo límite. Lo mismo sucede en los casos 3 y 4, en el período 6. Sin embargo, en todos estos casos si se obtienen soluciones con AG.

En cada caso, el GAP obtenido con CPLEX se incrementa al aumentar los períodos considerados. Por ejemplo, en el caso 1, al considerar un período, se encuentra la solución óptima a los 6 segundos. Para este mismo caso, al considerar 6 períodos, se encuentra la mejor solución a los 3540 segundos, con un GAP de 87.19%, por lo tanto dado la magnitud del GAP, no es posible tener certeza que se ha llegado a una solución óptima. Esto muestra como aumenta la complejidad del problema al incrementarse los períodos, ya que aumenta la cantidad de cajas por período a palletizar. En relación a  $DIF_H$  de la heurística GA, se observa que, a medida que aumenta el número de períodos, éste se va haciendo negativo en los casos más complejos. Esto significa que con AG se obtienen mejores soluciones que con CPLEX en los casos más complejos, al aumentar el horizonte de planificación.

En cuanto al tiempo de ejecución, a partir de un horizonte de 2 períodos, CPLEX detiene la ejecución cuando se llega al tiempo límite TF. Por esta razón, a veces se obtienen soluciones que no son óptimas. En el caso de AG, el tiempo límite TF es menor a los 7 segundos en todos los casos, incluidas las instancias más complejas. Esto muestra la rapidez de la heurística implementada para alcanzar una buena solución.

Tal como se explicó anteriormente, a partir de  $t = 2$ , el GAP de CPLEX es mayor a 0% al llegar al tiempo límite TF, por lo tanto, no se puede garantizar que se haya llegado a una solución óptima. Para analizar en cuanto influye el tiempo límite de ejecución en la obtención de la solución óptima, se analizaron tres instancias, en las que se incrementó el tiempo límite de ejecución a 10800 segundos. La comparación de los resultados cuando se utilizan diferentes TF se presenta en la Tabla 4. En esta tabla se observa que la disminución del GAP y la mejora de la función objetivo son poco significativas. Además, las mejores soluciones encontradas ocurren casi al final del tiempo de ejecución TF.

Tabla 4. Comparación de resultados de CPLEX incrementando el tiempo de ejecución

Casos	Período	t = 3600 (seg)			t = 10800 (seg)		
		FO	GAP (%)	TAS (seg)	FO	GAP (%)	TAS (seg)
1	6	641	87.19	3540	636	86.24	10330
3	4	712	61.11	3600	685	59.29	10689
5	2	955	30.62	3150	952	30.23	10400

## 6. Conclusiones

En este trabajo se propone un modelo matemático de programación lineal entera que aborda una extensión del *Pallet Loading Problem*, ya que, además de evaluar las diferentes maneras de conformar un pallet considera que esta actividad se deba realizar en una determinada área. El modelo desarrollado es NP-hard y se puede utilizar para apoyar en las decisiones de palletizado a nivel operativo, considerando factores tales como: la estructura del área de palletizado, condiciones de salida de los pallets completos, conformación de pallets de un sólo producto o mixtos.

Se realizó una experimentación computacional donde se compararon los resultados obtenidos

con CPLEX y un algoritmo goloso. El algoritmo goloso obtiene mejores soluciones que con CPLEX en los casos más complejos, al aumentar el horizonte de planificación.

En el área de palletizado faltan desarrollar modelos de optimización que consideren restricciones técnicas propias de la industria, siendo esta investigación una contribución en este ámbito. Para futuras investigaciones, se recomienda considerar y evaluar otras estrategias para la carga de pallet, como, por ejemplo, conformar pallets mixtos con diferentes tamaños de cajas, para así minimizar el ítem de costos por cantidad de productos en pallets mixtos.

### Referencias

- Alvarez-Valdés, R., Parreño, F., y Tamarit, J. M.** (2005). A branch-and-cut algorithm for the pallet loading problem. *Computers & Operations Research*, 32(11):3007-3029.
- Arghavani, J. y Abdou, G.** (1996). 3d volumetric pallet-loading optimisation. *The International Journal of Advanced Manufacturing Technology*, 11(6):425-429.
- Chao, J., Chen, M., Deng, A., Miao, H., Newman, A., Tseng, S., y Yano, C. A.** (2005). Safe way designs mixed-product pallets to support just-in-time deliveries. *Interfaces*, 35(4):294-307.
- Chen, C., Sarin, S., y Ram, B.** (1991). The pallet packing problem for non-uniform box sizes. *The International Journal of Production Research*, 29(10):1963-1968.
- Dowland, K. A.** (1987). An exact algorithm for the pallet loading problem. *European Journal of Operational Research*, 31(1):78-84.
- Gilmore, P. y Gomory, R.** (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94-120.
- Hausman, W. H., Schwarz, L. B., y Graves, S. C.** (1976). Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6):629-638.
- Hodgson, T. J.** (1982). A combined approach to the pallet loading problem. *IIE Transactions*, 14(3):175-182.
- Junqueira, L., Morabito, R., y Sato Yamashita, D.** (2012). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research*, 39(1):74-85.
- Lai, K., Xue, J., y Zhang, G.** (2002). Layout design for a paper reel warehouse: A two-stage heuristic approach. *International Journal of Production Economics*, 75(3):231-243.
- Larson, T. N., March, H., y Kusiak, A.** (1997). A heuristic approach to warehouse layout with class-based storage. *IIE transactions*, 29(4):337-348.
- Lashkari, R., Balakrishnan, B., y Dutta, S.** (2002). A multi-objective model for the allocation of pallets in a flexible manufacturing cell. *International Journal of Industrial Engineering-Theory Applications and Practice*, 9(3):287-300.
- Le-Duc, T. y De Koster, R. B.** (2005). Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse. *International Journal of Production Research*, 43(17):3561-3581.
- Liu, F.-H.F. y Hsiao, C.-J.** (1997). A three-dimensional pallet loading method for single-size boxes. *Journal of the Operational Research Society*, 726-735.
- Martins, G. H. y Dell, R. F.** (2008). Solving the pallet loading problem. *European Journal of Operational Research*, 184(2):429-440.
- Petersen, C. G., Aase, G. R., y Heiser, D. R.** (2004). Improving order-picking performance through the implementation of class-based storage. *International Journal of Physical Distribution & Logistics Management*, 34(7):534-544.
- Ram, B.** (1992). The pallet loading problem: A survey. *International Journal of Production Economics*, 28(2):217-225.
- Terno, J., Scheithauer, G., SommerweiB, U., y Riehme, J.** (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, 123(2):372-381.
- Tsai, R. D., Malstrom, E. M., y Kuo, W.** (1993). Three dimensional palletization of mixed box sizes. *IIE transactions*, 25(4):64-75.
- Tsai, R. D., Malstrom, E. M., y Meeks, H. D.** (1988). A two-dimensional palletizing procedure for warehouse loading operations. *IIE transactions*, 20(4):418-425.